



Universidad
Zaragoza

TRABAJO FIN DE GRADO

ANÁLISIS TÉRMICO Y DE
POTENCIA DE
MULTIPROCESADORES CON DVFS

THERMAL AND POWER ANALYSIS
OF MULTICORES WITH DVFS

AUTOR

MARIUS SORIN CRISAN

DIRECTORES

JOSÉ LUIS BRIZ VELASCO
VÍCTOR VIÑALS YÚFERA

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2019

RESUMEN

El consumo de los dispositivos electrónicos es un tema de investigación muy importante. El interés generado por este campo se debe principalmente a dos factores: el incumplimiento de la ley de Moore en los procesadores más recientes y la demanda de una mayor autonomía en dispositivos portátiles como el móvil. A pesar de ello, la existencia de caracterizaciones energéticas en procesadores es muy escasa, debido en parte a que los fabricantes apenas ofrecen documentación.

El trabajo comienza con un estudio de los fundamentos del control de potencia en procesadores. A continuación, se describe y se realiza la instrumentación del entorno experimental para modelar la caracterización térmica y energética de la Raspberry Pi 3B, una plataforma simple y barata. También se estudian los planos de control de voltajes y frecuencias de la plataforma junto con la gestión de potencia y rendimiento. Esta parte termina con el análisis del comportamiento térmico y energético de la Raspberry Pi. En la segunda parte del proyecto se analiza el mercado de las placas base y se configura un equipo que permita realizar un ajuste más fino del control de potencia y rendimiento que el hecho en la Raspberry Pi. Éste está compuesto por un procesador *Intel Core i7-7800X* y la placa base *Asus Rampage VI Extreme Omega*. Tras esto, se estudia el control de potencia y rendimiento de la plataforma. Por último, se instrumenta un entorno experimental a partir del cual se modela el comportamiento energético y térmico del equipo adquirido.

Una de las aportaciones del proyecto es la instrumentación de un entorno experimental para modelar el comportamiento energético de las dos plataformas con la posibilidad de adaptarlo a otros equipos similares. También se aporta la documentación del control de potencia y rendimiento más la distribución de alimentación y señales de reloj en cada plataforma. Por último, se analizan los modelos térmicos y energéticos de cada equipo junto con las limitaciones de cada plataforma.

ÍNDICE GENERAL

1	INTRODUCCIÓN	1
1.1	Fundamento del problema	1
1.2	Objetivos	2
1.3	Alcance	2
1.4	Planificación	2
1.5	Descripción del documento	2
2	FUNDAMENTOS	5
2.1	Potencia disipada	5
2.1.1	Potencia estática	5
2.1.2	Potencia dinámica	6
2.1.3	Tipos de transistores	8
2.2	Técnicas de ahorro energético	9
2.3	Trabajos relacionados	11
3	PLATAFORMA RASPBERRY	13
3.1	Dominios de frecuencia	14
3.2	Distribución eléctrica	16
3.3	Control de potencia y rendimiento	18
4	EXPERIMENTACIÓN	21
4.1	Medición	21
4.2	Software y metodología	22
4.3	Experimento 1: potencia dinámica y frecuencia	23
4.3.1	Descripción	23
4.3.2	Resultados	24
4.4	Experimento 2: potencia y voltaje	25
4.4.1	Descripción	25
4.4.2	Resultados	25
4.5	Conclusiones	26
5	PLATAFORMA ASUS	27
5.1	Placa base y componentes auxiliares	27
5.2	Procesador	29
5.3	Dominios de frecuencia	29
5.4	Distribución eléctrica	30
5.4.1	Serial Voltage Identification (SVID)	34
5.4.2	Calibración de la línea de carga	36
5.5	Control de potencia y rendimiento	36
5.5.1	Sensor Térmico Digital	38
5.5.2	Enhanced Intel SpeedStep Technology	39
5.5.3	Intel Speed Shift Technology	40
5.5.4	Intel Turbo Boost Max Technology 3.0	41
6	EXPERIMENTACIÓN Y RESULTADOS	43
6.1	Medición	43
6.2	Software	44

6.2.1	Tortura del procesador	45
6.2.2	Recolección de datos	46
6.3	Experimento 1: potencia dinámica y frecuencia	47
6.3.1	Descripción	47
6.3.2	Resultados	48
6.4	Experimento 2: potencia y voltaje	49
6.4.1	Descripción	49
6.4.2	Resultados	51
6.5	Experimento 3: potencia estática y temperatura	53
6.5.1	Descripción	53
6.5.2	Resultados	53
6.5.3	Conclusiones	53
7	CONSIDERACIONES FINALES	55
7.1	Conclusiones	55
7.2	Trabajo futuro	55
A	ANEXOS	57
A.1	Diagrama Gantt del proyecto	57
A.2	Velocidad de conmutación MOSFET	58
A.3	Gestión de energía y rendimiento en Linux	60
A.4	Advanced Configuration and Power Interface	63
A.5	Elección programa para estresar el procesador	65
A.6	BIOS	66
A.7	Principales características del procesador Intel utilizado	68
A.8	Fuente de alimentación	69
A.9	Refrigeración	69
	BIBLIOGRAFÍA	71

ÍNDICE DE FIGURAS

Figura 2.1	Transistor MOSFET 2D de tipo P.	6
Figura 2.2	Puerta NOT con dos MOSFET	7
Figura 2.3	Transistor MOSFET 3D o tri-gate	9
Figura 2.4	Vista microscópica del transistor 2D vs 3D	9
Figura 3.1	Diagrama de bloques de la Raspberry Pi	13
Figura 3.2	Vista frontal de la Raspberry Pi	14
Figura 3.3	Vista trasera de la Raspberry Pi	14
Figura 3.4	Dispadores en la Raspberry Pi	14
Figura 3.5	Oscilador de la Raspberry Pi	15
Figura 3.6	Árbol de señales de reloj de la Raspberry	16
Figura 3.7	Distribución eléctrica en la Raspberry Pi	17
Figura 3.8	DVFS en la Raspberry Pi 3B	18
Figura 4.1	Analizador de potencia Newtons4th PPA520	21
Figura 4.2	Medición de potencia en la Raspberry Pi 3B	22
Figura 4.3	Relación entre potencia y frecuencia en la Raspberry	24
Figura 4.4	Variación de la potencia al cambiar el voltaje	25
Figura 5.1	Diagrama de componentes de la placa Asus	27
Figura 5.2	Asus Rog Ramapge VI Extreme Omega	28
Figura 5.3	Distribución interna del Intel Core i7-7800X	30
Figura 5.4	Árbol de señales de reloj en la placa Asus	31
Figura 5.5	Distribución eléctrica en la placa Asus	32
Figura 5.6	VRM en la placa Asus	33
Figura 5.7	Conexiones del protocolo de comunicación SVID.	34
Figura 5.8	Variación del voltaje usando SVID	36
Figura 5.9	Variación del voltaje usando LLC	37
Figura 5.10	Cambio de frecuencia usando EIST	39
Figura 5.11	Comportamiento dinámico del Intel Turbo	41
Figura 6.1	Conexiónado eléctrico para la medición de potencia en la placa Asus.	44
Figura 6.2	Conexión en la borna repartidora	45
Figura 6.3	Entorno experimental de la plataforma Asus	46
Figura 6.4	Flujo de datos en experimentación - Asus	47
Figura 6.5	Dispersión de las medidas	48
Figura 6.6	Variación de la potencia con la frecuencia	49
Figura 6.7	Comparación potencia dinámica entre Skylake, Ivy Bridge y Sandy Bridge	50
Figura 6.8	Potencia estática y dinámica al variar voltaje	51
Figura 6.9	Potencia vs. temperatura	54

Figura A.1	Diagrama de Gantt	57
Figura A.2	Tiempo de conmutación en un MOSFET	58
Figura A.3	Zonas de funcionamiento de un MOSFET	58
Figura A.4	Esquema puerta NOT con dos MOSFET	59
Figura A.5	Diagrama de bloques de CPUFreq	62
Figura A.6	Componentes principales de ACPI	63
Figura A.7	Estados ACPI y sus relaciones	65
Figura A.8	Potencia consumida por programas de estrés	66

ÍNDICE DE CUADROS

Cuadro 1.1	Horas dedicadas	3
Cuadro 5.1	Tensiones máximas y mínimas permitidas	34
Cuadro 5.2	Rango de voltajes de funcionamiento de los raíles de alimentación familia Skylake-X	35

ACRÓNIMOS

TFG	Trabajo Fin de Grado
DVFS	Dynamic Voltage Frequency Scaling
IoT	Internet of Things
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistors
GPU	Graphics Processing Unit
AVS	Adaptive Voltage Scaling
BIOS	Basic Input/Output System
DRAM	Dynamic Random Access Memory
SoC	System on Chip
CPU	Central Processing Unit
SO	Sistema Operativo
ISA	Instruction Set Architecture

GPIO	General Purpose Input Output
PLL	Phase-Locked Loop
I ² C	Inter-Integrated Circuit
CPUFreq	CPU Frequency Scaling
NVMe	Non-Volatile Memory Express
SATA	Serial Advanced Technology Attachment
DIMM	Dual In-line Memory Module
RAID	Redundant Array of Independent Disks
SSD	Solid-State Drive
PSU	Power Supply Unit
PCIe	Peripheral Component Interconnect Express
TPU	Turbo Process Unit
PMIC	Power Management Integrated Circuit
SVID	Serial Voltage Identification
LLC	Load-Line Calibration
VRM	Voltage Regulator Module
PWM	Pulse-Width Modulation
EIST	Enhanced Intel SpeedStep Technology
RAPL	Running Average Power Limit
DTS	Digital Thermal Sensor
TCC	Thermal Control Circuit
PCU	Unidad de Control de Potencias
TDP	Thermal Design Power
PECI	Platform Environment Control Interface
PMA	Power Management Agents
PCH	Platform Controller Hub
LCC	Low Core Count
HCC	High Core Count
XCC	eXtreme Core Count

BCLK Base Clock

DMI Direct Media Interface

ACPI Advanced Configuration and Power Interface

MSR Model Specific Register

ESST Intel Speed Shift Technology

ACPI Advanced Configuration and Power Interface

OSPM Operating System-directed configuration and Power
 Management

AML ACPI Machine Language

AVX-512 512-bit Advanced Vector Extensions SIMD

IPS Instrucciones por segundo

INTRODUCCIÓN

1.1 FUNDAMENTO DEL PROBLEMA

La potencia consumida por un procesador se compone de la potencia estática, fundamentalmente debida a corrientes de fuga, y de la potencia dinámica consumida por la actividad de conmutación de los transistores. La potencia estática depende de la temperatura de una forma en general no lineal, característica de cada circuito. La potencia dinámica es proporcional a la frecuencia y al cuadrado de la tensión de alimentación, dependiendo de una constante de proporcionalidad también característica de cada circuito. Aunque existen modelos para estimar la potencia estática y la constante de proporcionalidad de la potencia dinámica, solamente la experimentación puede ofrecer un dato fiable para un procesador dado. Esta experimentación se basa en la medida directa de la potencia consumida por el procesador mientras ejecuta un software intensivo en cálculo, con una utilización irrelevante de la memoria dinámica y del subsistema de entradas - salidas. Realizando experimentos en los que se fijan unos valores mientras se varían otros, del conjunto de variables que influyen en la potencia (temperatura, voltaje, frecuencia), se pueden estimar la potencia estática y el modelo de variación de potencia dinámica respecto al voltaje y frecuencia.

La realización de estas medidas no es trivial. No puede realizarse en cualquier tipo de sistema ni con cualquier tipo de procesador. El BIOS de la placa, el *firmware* de gestión de potencia integrado en los reguladores de tensión de la misma y en el propio procesador, o las interacciones de los drivers de distribución de alimentación y señales de reloj entre otros factores, complican la toma de medidas. A estas dificultades hay que añadir la del control de temperatura, y la escasa documentación pública sobre los elementos que hemos citado.

Sin embargo, sólo un trabajo de estas características permite adentrarse en los detalles de gestión de potencia de un microprocesador comercial moderno. Tanto el modelo de potencia como el estudio de los detalles del sistema de distribución y gestión de potencia y frecuencia son imprescindibles para dar solidez a la investigación en diferentes campos. Por ejemplo, sin un modelo adecuado de potencia no es posible diseñar y evaluar de forma fiable mecanismos de control de temperatura y minimización del consumo, un tópico de tremenda actualidad en Arquitectura de Computadores, prácticamente en todos los segmentos (servidores, computación personal, y sistemas empujados tiempo real). El solo hecho de elaborar una metodología adecuada

y fiable para estimar modelos de potencia supone una aportación importante.

1.2 OBJETIVOS

El objetivo general de este Trabajo Fin de Grado (TFG) es la selección, preparación e instrumentación de un sistema que permita investigar el modelo de potencia de un microprocesador, estableciendo una metodología apropiada, y proporcionando un conocimiento lo más preciso posible de los mecanismos en placa base y procesador que intervienen en la distribución y regulación de potencia.

Los objetivos específicos son:

- Realizar un primer análisis sobre un sistema sencillo de disponibilidad inmediata, como una Raspberry Pi.
- Selección, adquisición, montaje y configuración del sistema (placa base, fuente, microprocesador, elementos de control de temperatura, vatímetro).
- Estudiar, comprender y documentar la gestión de potencia de la placa y procesador utilizados.
- Seleccionar programas para estresar el procesador y herramientas para recolectar los datos.
- Realizar experimentos y analizar los resultados obtenidos

1.3 ALCANCE

Los resultados obtenidos en el desarrollo del proyecto, recogidos en esta memoria y sus anexos, constituyen el primer paso en la estimación de un modelo de potencia para un microprocesador comercial actual, que será incorporado a diferentes vías de investigación del Grupo de Arquitectura de Computadores de la Universidad de Zaragoza (GaZ).

1.4 PLANIFICACIÓN

El diagrama de Gantt del proyecto está en el Anexo [A.1](#) y en el Cuadro [1.1](#) se muestran las horas dedicadas.

1.5 DESCRIPCIÓN DEL DOCUMENTO

El resto de esta memoria se organiza como sigue. El [Capítulo 2](#) proporciona información básica sobre el modelo de potencia de un procesador y los elementos y conceptos en los que se basa. El [Capítulo 3](#) introduce las características de la primera plataforma estudiada

TAREA	HORAS
1. Estudio de trabajos previos	7
2. Estudio de fundamentos	22
3. Configuración del entorno experimental Raspberry	5
4. Documentación de la plataforma Raspberry	33
5. Medición y análisis de resultados Raspberry	20
6. Identificación y configuración de la plataforma Asus	36
7. Configuración equipo de medida	17
8. Documentación de la placa base Asus	67
9. Documentación del procesador Intel	98
10. Configuración del entorno experimental Asus	36
11. Experimentos plataforma Asus	58
11. Escritura memoria y documentación	90
Horas totales:	489

Cuadro 1.1: Horas dedicadas

(Raspberry Pi) y sus detalles en cuanto gestión de energía, temperatura y frecuencia. El [Capítulo 4](#) presenta la instrumentación y experimentos realizados sobre la Raspberry Pi. El [Capítulo 5](#) explica la arquitectura de gestión de potencia y *clocking* de la placa Asus y del procesador Intel Skylake montado. Los resultados experimentales sobre este último sistema se presentan en el [Capítulo 6](#). Finalmente, el [Capítulo 7](#) recapitula el trabajo realizado y establece sus líneas de continuación. Los anexos recogen documentación y material complementario.

FUNDAMENTOS

En este capítulo se explica la potencia disipada por un procesador y algunas técnicas utilizadas para reducir su consumo. También se analiza el estado del arte en cuestiones de caracterización térmica y energética en procesadores.

2.1 POTENCIA DISIPADA

Dada su naturaleza, toda la potencia consumida por un procesador se transforma en calor. Debido a que este no deja de ser un gran circuito de conmutación, se puede dividir en dos tipos, potencia estática y dinámica:

$$P_{total} = P_{est} + P_{din} \quad (2.1)$$

2.1.1 Potencia estática

El consumo estático se debe a las corrientes de fuga producidas en los transistores de efecto de campo (*Metal-oxide-semiconductor Field-effect transistor* o **MOSFET**) cuando no están conmutando (Ec. 2.2). Se diferencian dos corrientes principales: la corriente subumbral (I_{sub}) y la corriente de la puerta de óxido (I_{ox}).

La primera aparece al aplicar una diferencia de potencial entre la fuente y el drenador del transistor. En el mundo macroscópico, no debería haber ninguna pérdida porque el sustrato es un semiconductor y el transistor debería comportarse como un condensador. Pero, debido a la pequeña distancia que separan ambos extremos, el transistor deja de comportarse como una carga capacitiva y se genera un pequeño canal entre la fuente y el drenador a través del cual pasa una pequeña corriente. El efecto que produce la segunda corriente es el mismo, pero ahora esta corriente se establece entre la puerta y el sustrato debido al grosor del óxido que separa ambas partes (Fig. 2.1).

$$P_{est} = VI_{fuga} = V(I_{sub} + I_{ox}) \quad (2.2)$$

El porcentaje de potencia estática respecto a la potencia total disipada por un procesador se incrementa al disminuir la escala de integración. Años atrás, con escalas de integración superiores a 32 nm y técnicas muy efectivas de reducción de corrientes de fuga, la

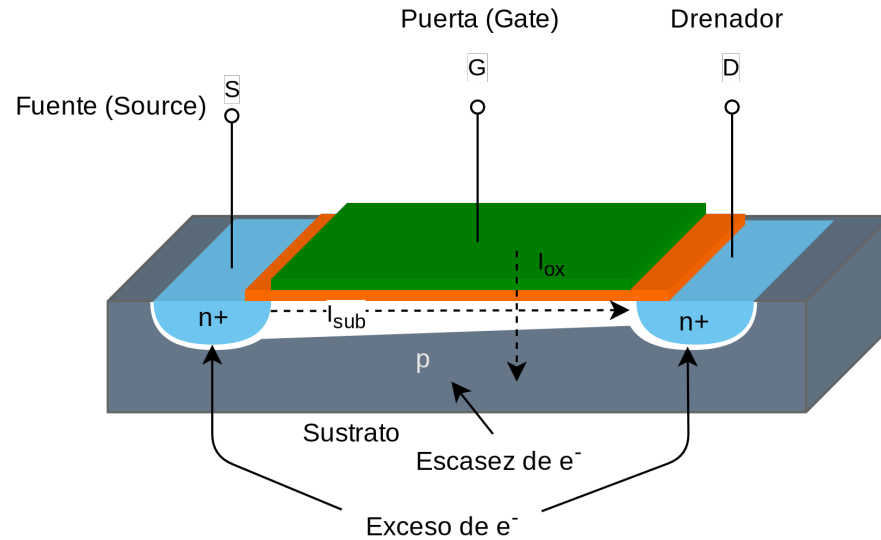


Figura 2.1: Transistor MOSFET 2D de tipo P.

potencia estática no era una pequeña parte de la potencia total. Actualmente es más relevante dado que su valor ha incrementado al reducir la escala de integración por debajo de los 20nm.

2.1.2 Potencia dinámica

La potencia dinámica se produce durante la conmutación de un transistor. Para entender como se genera esta potencia se explica el funcionamiento de una puerta NOT (Fig. 2.2) construida con dos transistores MOSFET. Cuando la tensión de entrada es menor que la tensión de umbral ($V_i = 0V < V_{th}$) se cumple que $V_o = V_{DD}$ porque el transistor NMOS no conduce ($V_{SG1} = V_i = 0V$) mientras que el transistor PMOS sí conduce ($V_{SG2} = V_{DD}$). En caso contrario, ($V_i = V_{DD}$), conducirá el NMOS y no el PMOS, y por tanto, $V_o = 0V$.

En la salida V_o hay conectado un condensador que representa la capacidad equivalente de las puertas conectadas a la puerta NOT. La conmutación de esta capacidad ($0 \rightarrow 1$ o $1 \rightarrow 0$) viene determinada por la actividad del procesador. Además, los valores concretos de capacidad varían de un modelo de procesador a otro y también entre los del mismo modelo debido a las variaciones en el proceso de fabricación (*process variation*) explicadas en la Sec. 2.2. Durante la transición de un valor lógico a otro hay que descargar o cargar el condensador, por lo que el tiempo necesario para que el transistor alcance un nuevo valor correcto varía en función de la capacidad del condensador y la diferencia de voltaje entre el drenador y la fuente (detalles en el Anexo A.2). En resumen, la potencia dinámica está definida por la Ec. 2.3, donde C_L representa la capacidad equivalente del chip completo, f la frecuencia de conmutación, V el voltaje y A la actividad o porcentaje de conmutaciones simultáneas del procesador.

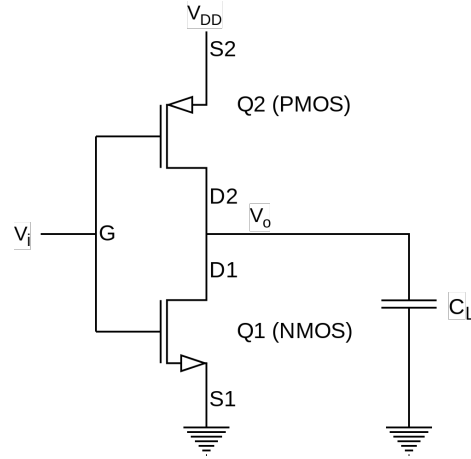


Figura 2.2: Esquema de una puerta NOT con dos MOSFET. Los tres terminales de cada transistor se llaman puerta, drenador y fuente: G, D y S, respectivamente.

En la práctica, la actividad se suele aproximar a 0,5 para el procesador, pero puede bajar mucho para las memorias cache en chip de último nivel.

$$P_{din} = A \cdot C_L f V^2 \quad (2.3)$$

El escalado de frecuencia máxima es lineal con la tensión [31, 40]. Por tanto, al reducir la frecuencia lo hace también la tensión (Ec. 2.4), pudiéndose aproximar la potencia total a la Ec. 2.5. Es decir, la potencia total consumida por un procesador varía cúbicamente con la frecuencia.

$$f_{max} \propto \frac{(V - V_{th})^2}{V} \quad (2.4)$$

$$P_{total} \simeq AC_L f^3 + P_{est} \quad (2.5)$$

Sin embargo, hay que tener en cuenta que al reducir el voltaje (V_{DD}) también hay que reducir la tensión umbral (V_{th}), lo que conlleva un aumento exponencial de la potencia estática, ya que la intensidad de fuga aumenta (Ec. 2.6). Además, la corriente de fuga aumenta con la temperatura:

$$I_{fugas} \propto e^{(-qV_{th}/kT)} \quad (2.6)$$

Teniendo en cuenta estas aproximaciones, se plantea la siguiente hipótesis: *Ejecutando en dos procesadores una tarea X totalmente paralelizable a mitad de frecuencia (y voltaje) se consume un cuarto de la energía consumida en una ejecución secuencial (E_1)* (Ec. 2.7).

$$E_1 = P_1 t$$

$$E_2 = 2 \frac{P_1}{8} t = \frac{E_1}{4} \quad (2.7)$$

P_1 es la potencia consumida en la versión secuencial y t el tiempo que tarda en terminar (Ec. 2.7).

2.1.3 Tipos de transistores

En el año 2011 Intel® lanzaba al mercado el primer procesador de propósito general construido con transistores 3D [29]. Esto supuso un cambio en el proceso de fabricación de los procesadores, aunque el principio de funcionamiento seguía siendo el mismo que el de los transistores 2D.

2.1.3.1 Transistor 2D

El transistor plano o 2D tiene la fuente y el drenador incrustados en el sustrato. Además, el tipo de dopaje (tipo n o p) debe ser distinto entre el sustrato y los otros dos terminales (Fig. 2.1). La puerta está separada del sustrato por una pequeña capa de óxido.

El principal problema de este transistor es el gran incremento de las corrientes de fugas al disminuir el tamaño del transistor. Por ejemplo, la *corriente subumbral* (I_{sub}) se genera porque la superficie de la fuente y el drenador son muy grandes y se crea una gran carga capacitiva. Por tanto, cuando ambos terminales están separados por una distancia tan pequeña, incrementa la corriente subumbral ya que la fuerza del campo eléctrico es mayor. Si la diferencia de potencial aplicada en los dos terminales fuera muy alta puede que la puerta no sea capaz de seguir controlando el funcionamiento del transistor ya que la *corriente subumbral* sería muy cercana a la corriente de conducción del transistor en estado de saturación. Estas pérdidas incrementan mucho con escalas de integración inferiores a 32nm.

2.1.3.2 Transistor 3D

La diferencia del transistor 3D o *tri-gate* con los 2D reside en su geometría. En los transistores 3D, fuente, puerta y drenador sobresalen de la superficie del sustrato. La puerta es un escalón perpendicular que envuelve al canal que forma la fuente y el drenador (Fig. 2.3). Esto hace que la puerta sea capaz de controlar el transistor con una tensión umbral menor. La reducción del voltaje umbral se debe a que la anchura del canal es menor, lo que hace que su carga capacitiva disminuya.

La intensidad máxima del transistor se regula con la altura del canal o añadiendo más canales en paralelo. Las razones que hacen que estos transistores sean los más usados en circuitos con una escala de integración inferior a 20 nm son la disminución de la potencia estática

y de la latencia de conmutación junto al incremento de la corriente conducida en el estado de saturación [43] [45].

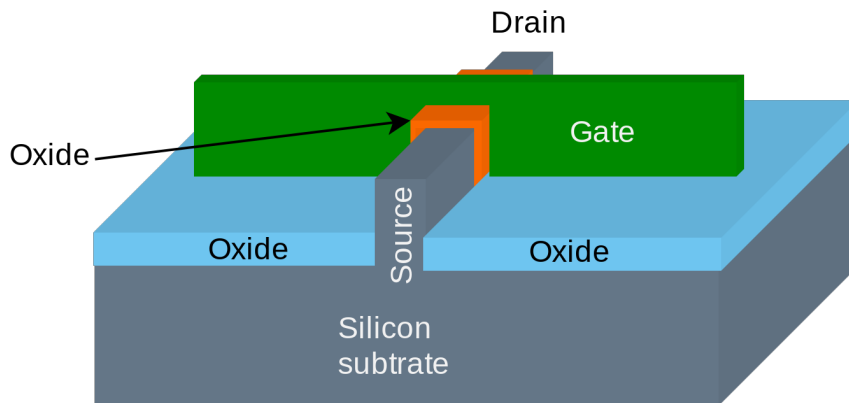


Figura 2.3: Transistor MOSFET 3D o tri-gate

En la Fig. 2.4 se muestra una vista microscópica de un transistor plano de Intel® de 32nm y otro 3D de 22nm.

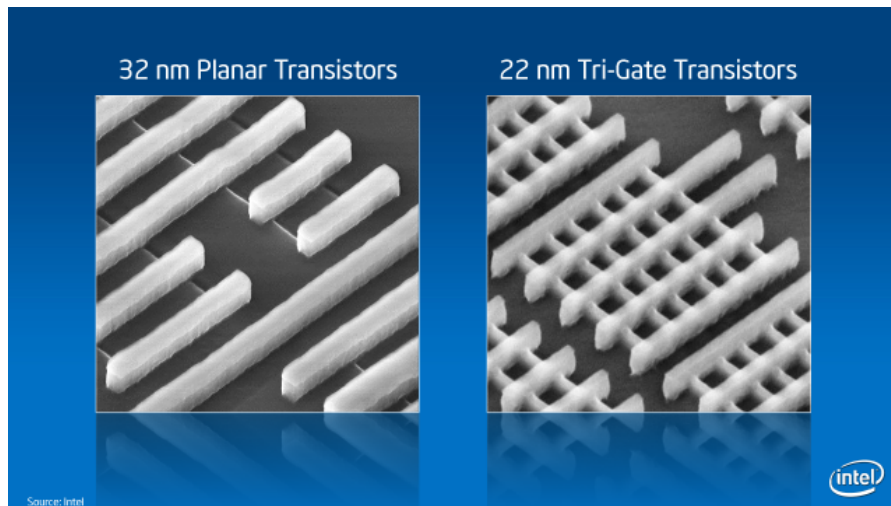


Figura 2.4: Vista microscópica de un transistor 2D (izquierda) y otro 3D (derecha). Fuente: [29].

2.2 TÉCNICAS DE AHORRO ENERGÉTICO

En esta sección se presentan cinco de las técnicas más usadas para *thermal throttling* en los procesadores actuales.

2.2.0.1 Instrucción de bajo consumo

Esta técnica consiste en insertar instrucciones de bajo consumo en el *pipeline* de ejecución del procesador, por ejemplo *nops*. Estas instrucciones reducen el consumo de ciertas zonas del procesador como las unidades vectoriales, de coma flotante o de enteros ya que no las usan.

Thermal throttling o ahogamiento térmico: estado térmico del procesador en el que se alcanza una temperatura demasiado alta y es necesario aplicar alguna técnica para rebajarla.

2.2.0.2 Escalado dinámico de voltaje y frecuencia

El escalado dinámico de voltaje y frecuencia o *Dynamic Voltage and Frequency Scaling* (DVFS) es un mecanismo de control de potencia y rendimiento que consiste en cambiar la frecuencia y/o el voltaje de ciertas zonas del procesador como la CPU, cache, controladores de memoria, GPU, etc.

En la mayoría de los equipos en los que está disponible, el Sistema Operativo (SO) o el propio procesador se encarga de modificar frecuencia y voltaje basándose en parámetros como la temperatura o la carga del procesador. En equipos destinados a *overclocking* es común encontrarse con parámetros en el BIOS o en programas del SO que permiten al usuario modificar dichas variables. Se explicará más en detalle en los Cap. 3 y 5.

El escalado de voltaje adaptativo (*Adaptive Voltage Scaling*, AVS) es un caso específico de DVFS que solo modifica la tensión suministrada a un dominio del procesador para ahorrar energía.

No todos los procesadores del mismo modelo pueden alcanzar las mismas configuraciones de DVFS debido a las variaciones en el proceso de fabricación (*Process Variation*) del chip, producidas por factores incontrolables, que impiden que todos los procesadores rindan igual. Se pueden distinguir tres tipos de transistores según esta característica:

- Débil (*weak*): necesitan un voltaje más alto que gran parte de la población analizada para poder operar a la frecuencia nominal. Este tipo de dispositivos tienen una potencia estática menor que el resto de dispositivos de la misma población.
- Nominal: la tensión necesaria para que operen a la frecuencia intermedia es la nominal.
- Fuerte (*strong*): son capaces de operar a la frecuencia estándar con una tensión menor a la utilizada por la mayoría de transistores. A cambio, su potencia estática suele mayor que en los otros dos casos [14].

2.2.0.3 Media-Frecuencia

Esta técnica consiste en usar tanto el flanco de subida del reloj como el de bajada. Con esto se puede reducir la frecuencia del árbol de relojes a la mitad. A pesar de reducir la potencia dinámica consumida por el árbol de relojes, esta técnica no es muy utilizada debido a la complejidad añadida en el control de los registros.

2.2.0.4 Clock gating

Con esta técnica se deshabilita la señal de reloj que llega a ciertas zonas del procesador. La potencia dinámica se reduce, ya que la variable A de la Ec. 2.3 disminuye. La implementación de esta técnica

Overclocking y underclocking son técnicas consistentes en cambiar la frecuencia de un procesador respectivamente por encima o por debajo de su máxima o mínima frecuencia operativa estándar. El overclocking es frecuentemente utilizado por jugadores de videojuegos. Ambas técnicas especialmente la primera pueden dañar la placa o el procesador. Existen placas especialmente diseñadas para minimizar este riesgo.

requiere un circuito complejo para no alterar el resto de relojes al apagar el de una parte del procesador [40].

2.2.0.5 *Power gating*

Esta técnica consiste en cortar la alimentación de una zona del procesador. Es la forma en la que más energía se ahorra, aunque la disponibilidad del dispositivo en cuestión disminuye debido a que el tiempo necesario para que vuelva a funcionar es mayor que en el resto de casos.

2.3 TRABAJOS RELACIONADOS

En el últimos años se ha incrementado el interés por el control de potencia y rendimiento de los sistemas electrónicos, probablemente debido al auge del *Cloud*, *IoT* o de dispositivos como el teléfono móvil. Además, la escala de integración de los componentes electrónicos está alcanzando unos límites donde la potencia estática se incrementa de manera desmesurada.

El estudio que más se ajusta a los experimentos que se pretende realizar en este TFG es el publicado en el *Blog de Henry* [21]. En él se realiza una comparación energética entre un procesador Intel® Core i5 de la generación *Sandy Bridge* (32nm) y otro de la generación *Ivy Bridge*. El primero está construido con transistores 2D de 32nm y el segundo con transistores 3D de 22nm. En el primer experimento se fija un voltaje y se varía la temperatura mientras se registra la potencia total consumida. El experimento se repite dos veces a diferentes frecuencias. A partir de estos resultados se obtiene una gráfica con la variación de la potencia estática respecto a la temperatura y otra con la relación entre potencia dinámica y temperatura. En el segundo experimento se varía solo el voltaje para obtener el comportamiento de ambas potencias con el voltaje. Por último, se obtienen todos los puntos de operación (voltaje y frecuencia) de cada procesador.

También hay estudios de la potencia consumida en centros de datos. Por ejemplo, Khan et al. [32, 33] determina la precisión de los contadores *RAPL* (*Running Average Power Limit*) de Intel® en un centro de datos con más de 900 nodos.

Otro artículo relacionado es [38] en el que se razona la implicación que tiene el voltaje subumbral y el voltaje de alimentación usado por un transistor. Con ello, se implementa un controlador de potencia para multiprocesadores que consiste en ajustar la tensión para conseguir una mejor eficiencia energética del sistema. En los experimentos presentados no se realiza una caracterización térmica y energética del sistema obteniendo la potencia dinámica y estática por separado. Además, para los resultados finales de eficiencia energética se tiene en cuenta el consumo total del sistema y por tanto, en las medidas

obtenidas al estresar el procesador se produce un error en la medición al no descontar la energía consumida por el resto de componentes.

En el otro extremo, se encuentran las competiciones de *overclocking* donde se crean ligas según las capacidades de los equipos utilizados por los participantes [24]. La posición en estos rankings se establece según la puntuación que obtiene su equipo al ejecutar un determinado test. Como consecuencia, hay muchos foros dedicados al *overclocking* y también muchos blogs que tienen guías para realizar *overclocking* o revisiones sobre placas base y procesadores. Por ejemplo, en el blog xDevs [49] se explica como realizar *overclocking* a una Raspberry y unas modificaciones para conseguir mejores resultados, pero no se proporciona una caracterización de la potencia consumida.

Respecto al segundo equipo utilizado en este trabajo (Sec. 5), y otros similares, no se dispone de una documentación en la que se explique el control de potencia y rendimiento del equipo y un análisis de la potencia consumida por el procesador. Por ejemplo, en el portal de revisiones de *AnandTech* existe una publicación que proporciona una comparación de la potencia total consumida por un equipo con este procesador versus otros procesadores parecidos del mercado [52]. El problema de estos resultados es su poca precisión, ya que al medir la potencia tienen en cuenta todos componentes del equipo. Además, han sido obtenidos solo con un par fijo de frecuencia y voltaje por lo que no se puede obtener la potencia estática y dinámica del procesador.

PLATAFORMA RASPBERRY

En este capítulo se describe el equipo hardware empleado en la primera parte del trabajo, así como la razón por la que fue elegido. Además, se explica en detalle la gestión de la energía, temperatura y frecuencia que realiza este equipo.

El primer equipo sobre el cual se ha realizado una caracterización térmica y energética es una placa Raspberry Pi Model 3B [20] (Fig. 3.1). Su principal componente es el System on Chip (SoC) BCM2837 de BroadCom, compuesto por un procesador de cuatro núcleos construido con transistores 2D de 28nm [8]. La frecuencia base por defecto es de 1.2 GHz y tiene 40 pines con modo General Purpose Input Output (GPIO), una DRAM de 1 GB y una GPU integrada en el SoC. El SO se instala en una tarjeta microSD que posteriormente se inserta en la placa. Para este proyecto se ha utilizado la distribución Raspbian de Linux, con el kernel para 32 bits de la versión 4.14.

Raspbian, basada en Debian, es mantenida por la Raspberry Pi Foundation [18].

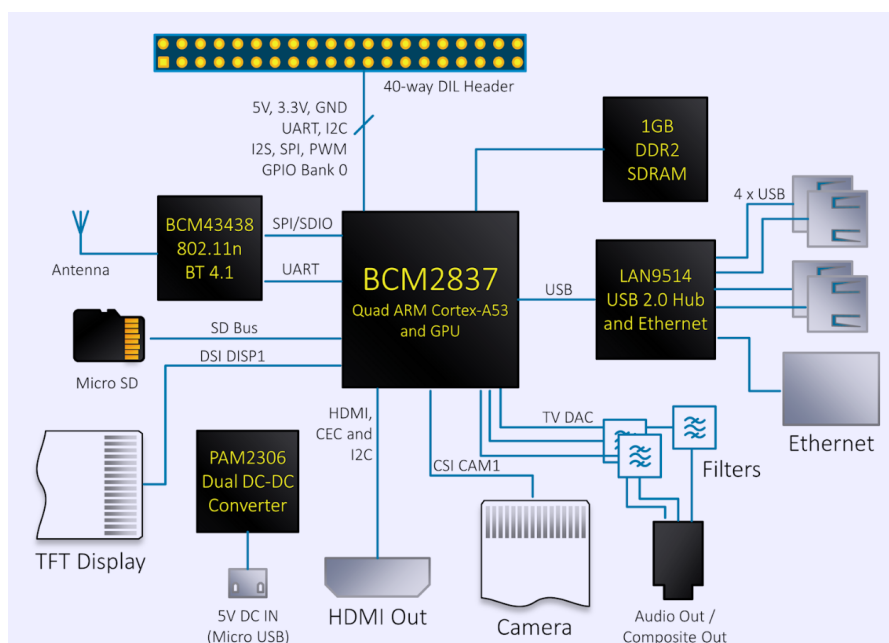


Figura 3.1: Diagrama de bloques de la Raspberry Pi 3 B
Fuente: Overclocking for Raspberry Pi 3 Model B [49]

La arquitectura de lenguaje máquina o Instruction Set Architecture (ISA) es ARMv8-A y soporta sistemas operativos tanto de 32 como de 64 bits. La comunicación con el dispositivo se puede realizar de forma remota mediante uno de los interfaces de red (WiFi o Ethernet), o de forma local mediante teclado, ratón y pantalla.

Al realizar los experimentos, la temperatura del procesador subía por encima de los 75°C dado que se había incrementado la frecuencia y el voltaje tal como se explica en el Cap. 4.3 Para bajar la temperatura del dispositivo se instalaron disipadores en el procesador, memoria RAM y el Hub que conecta los puertos USB y Ethernet (Fig. 3.4). También se conectó un pequeño ventilador de 5V a los pines 4 y 6 del interfaz GPIO (Fig. 4.2).

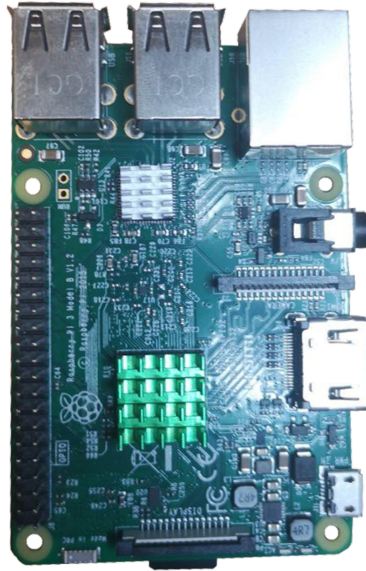


Figura 3.2: Vista frontal de la Raspberry Pi

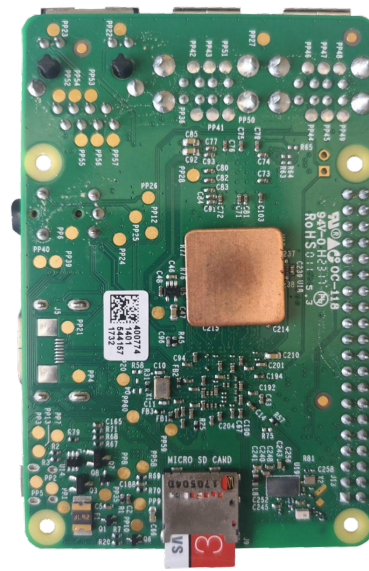


Figura 3.3: Vista trasera de la Raspberry Pi

Figura 3.4: Raspberry Pi 3B con disipadores instalados sobre el procesador (verde), USB/Ethernet hub (plata) y memoria RAM (naranja).

Se optó por trabajar con este equipo durante la primera parte del proyecto debido a dos razones. La primera fue la gran demora en la llegada del pedido con el equipo profesional destinado a *overclocking* (Cap. 5). La segunda, que toda la información relativa a la Raspberry está disponible por ser hardware libre (Raspberry Pi Foundation, [18]). Además, es un dispositivo sencillo y hay una guía oficial para hacer *overclocking* del mismo [17]. En este sentido, la Raspberry Pi resultó ideal para dar los primeros pasos del proyecto.

3.1 DOMINIOS DE FRECUENCIA

En la parte trasera de la Raspberry Pi 3B (Fig. 3.5) hay un oscilador de cristal que genera una señal de reloj de 19.2 MHz. Para variar su frecuencia se utilizan cinco Phase-Locked Loop (PLL)s independientes que multiplican la señal de entrada por un número entero [44]. La señal obtenida de cada PLL pasa por un divisor del que salen cuatro señales cuyas frecuencias son el resultado de dividir la frecuencia

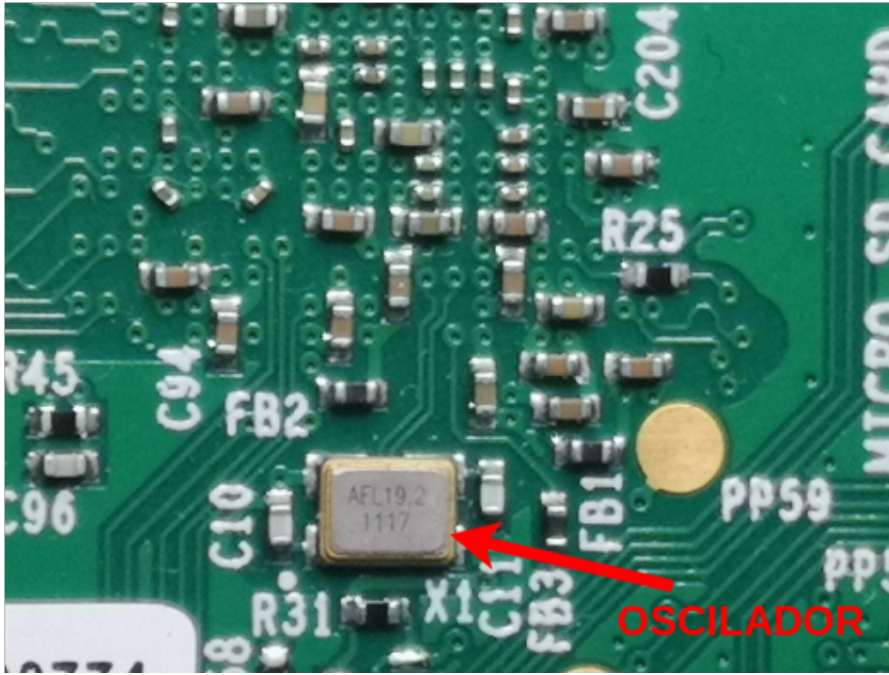


Figura 3.5: Oscilador de la parte trasera de la Raspberry Pi 3B

de entrada por un entero. De esta forma, modificando el valor de los registros hardware usados para los multiplicadores y divisores de los PLLs, se puede conseguir un amplio rango de frecuencias (Ec. 3.1), incluso que no sean múltiplo de 19.2 MHz [48].

$$F = 19,2\text{MHz} * \frac{\text{registro_PLL}}{\text{registro_divisor}} \quad (3.1)$$

En la figura 3.6 se muestra un diagrama con el árbol de señales de reloj utilizadas en la Raspberry Pi 3B. Este diagrama ha sido construido a partir de información obtenida de los fuentes del sistema operativo y varios blogs y foros [48].

Hemos podido determinar que la señal utilizada por el procesador es la que proviene del PLL C ya que sus cuatro señales descendientes varían al modificar la frecuencia del procesador. Algo parecido ocurre con el PLL H al conectar el cable HDMI (Fig. 3.6). En cambio, con la memoria RAM y la GPU no se han encontrado evidencias de que las conexiones sean como en la Fig. 3.6, aunque tendría que ser similar porque se puede cambiar la frecuencia de la memoria RAM y GPU de forma independiente.

Además, este SoC ofrece la posibilidad de usar los pines 7, 29 y 31 del interfaz GPIO como señal de reloj. Cada una de estas señales pueden usar directamente la señal generada por el oscilador (19.2 MHz) o una de las señales, configurables por software, que provienen de las salidas de los divisores A, C, D o H.

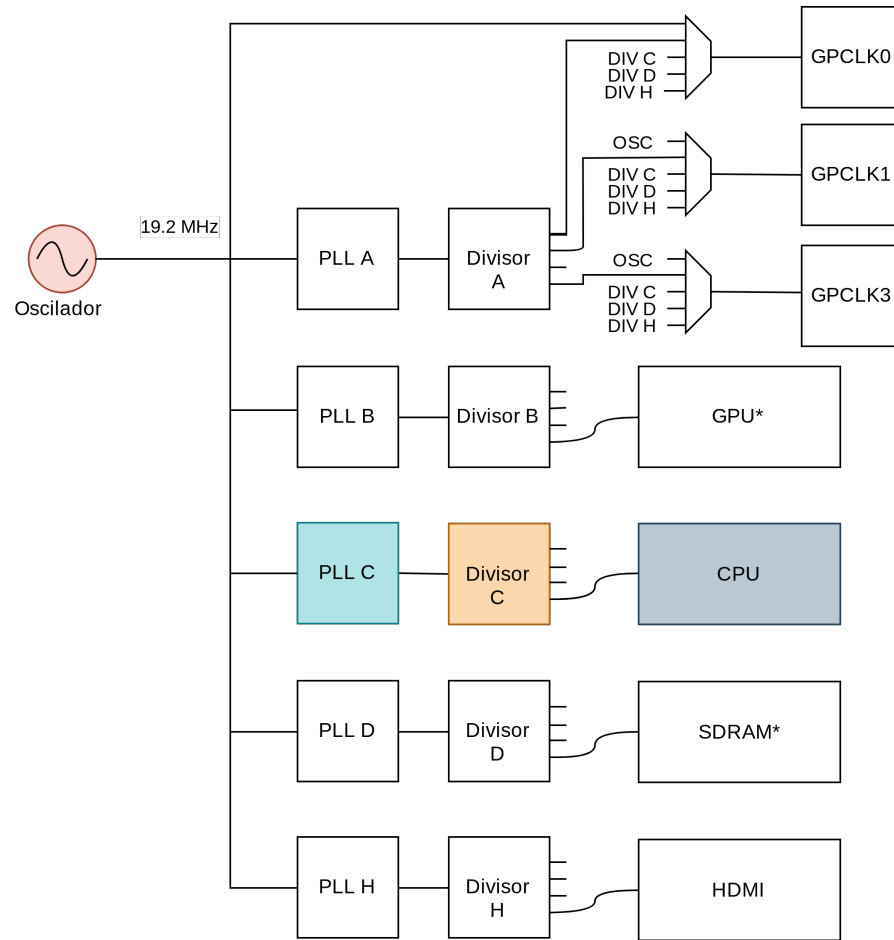


Figura 3.6: Diagrama con el árbol de señales de reloj (*clock tree*) de la Raspberry Pi 3B

3.2 DISTRIBUCIÓN ELÉCTRICA

En la Raspberry Pi 3B hay cuatro railes de voltaje: 5V, 3.3V, 1.8V y 1.2V. El raíl de 5V es el primero del circuito ya que la alimentación del equipo se realiza a través de un puerto *micro USB* a 5V. Este raíl consta de un fusible que limita la intensidad a 2.5A y un diodo (caída de tensión de 0.1V) construido con un transistor para permitir solo la corriente entrante al circuito. Estos 5V se utilizan para alimentar los pines 2 y 4 del interfaz **GPIO** más los puertos HDMI y USB. Para proteger los puertos USB y HDMI contra cortocircuitos o corrientes entrante hay otro diodo parecido al anterior.

En led que hay en una de las esquina de la placa se apaga cuando la tensión de entrada es inferior a los 4.7V. Este es controlado por un microprocesador *APX803* [26] que monitoriza el circuito eléctrico.

Los raíles de 3.3V y 1.8V se obtienen con un regulador conmutado doble [27]. El raíl de 3.3V se usa para alimentar el resto de pines de la interfaz **GPIO** y los puertos del audio, pantalla y cámara. En cambio,

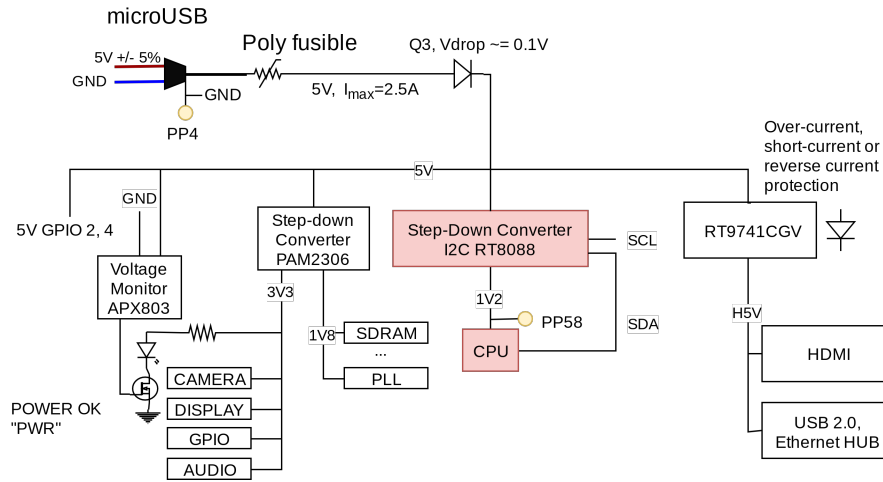


Figura 3.7: Distribución eléctrica en la Raspberry Pi 3B

módulos como la memoria RAM o los PLL de la Fig. 3.6 se alimentan desde el raíl de 1.8V.

En el arranque de la Raspberry, el último raíl en ser alimentado es el de 1.2V. La regulación del voltaje se realiza mediante un regulador conmutado RT8088A [11] dotado de comunicación serie. El procesador consta de un circuito integrado para el control de potencia (*Power Management Integrated Circuit* o *PMIC*) que establece la tensión adecuada en función de la temperatura y frecuencia del procesador tal como se explica en la Sec. 3.3. El voltaje establecido por el *PMIC* es enviado al regulador RT8088a mediante un bus I2C.

Este modelo se topa con varios problemas que fueron descubiertos durante la realización de este TFG. Por ejemplo, en los momentos de más carga, el sistema no es capaz de suministrar la potencia necesaria y algunos periféricos como el USB o lector microSD dejan de funcionar hasta un nuevo reinicio. De hecho, al cambiar la frecuencia del sistema este solicita demasiada potencia durante el arranque y, en consecuencia, el puerto SD deja de funcionar correctamente a mitad de una escritura y se corrompe el SO. Para evitar este problema, se cambió de una tarjeta SD *class 2* (2 MB/s escritura) a una *class 10* (10 MB/s escritura) con mejores características.

En la nueva Raspberry Pi 3B+ se integran todos estos dispositivos en un chip MXL7704 controlado mediante interfaz I2C [39]. Con esto, se evitan los problemas de alimentación del dispositivo usado en este TFG. Además, la utilización de este chip reduce el tiempo de respuesta de tal forma que, ante una variación de la carga del sistema, el tiempo que tarda en suministrarle la tensión necesaria al procesador es menor. En consecuencia, la eficiencia energética del dispositivo es mejor y se mejora la respuesta del sistema ante los picos de carga [1].

3.3 CONTROL DE POTENCIA Y RENDIMIENTO

La Raspberry Pi 3B utiliza la técnica de escalado dinámico de voltaje y frecuencia (DVFS) para conseguir un mayor ahorro energético. Este control lo realiza el subsistema *CPU Frequency Scaling* (CPUFreq) que consta de varias políticas de gestión de energía y rendimiento. Cada política responde a distintas heurísticas que en función de unos parámetros de entrada obtienen los valores adecuados de voltaje y frecuencia para que el equipo alcance un cierto balance entre rendimiento y consumo. La temperatura del chip o la carga de trabajo del SO son ejemplos de parámetros de entrada. Una vez obtenido el par de valores de voltaje y frecuencia, un driver se encarga de aplicar los cambios en el hardware.

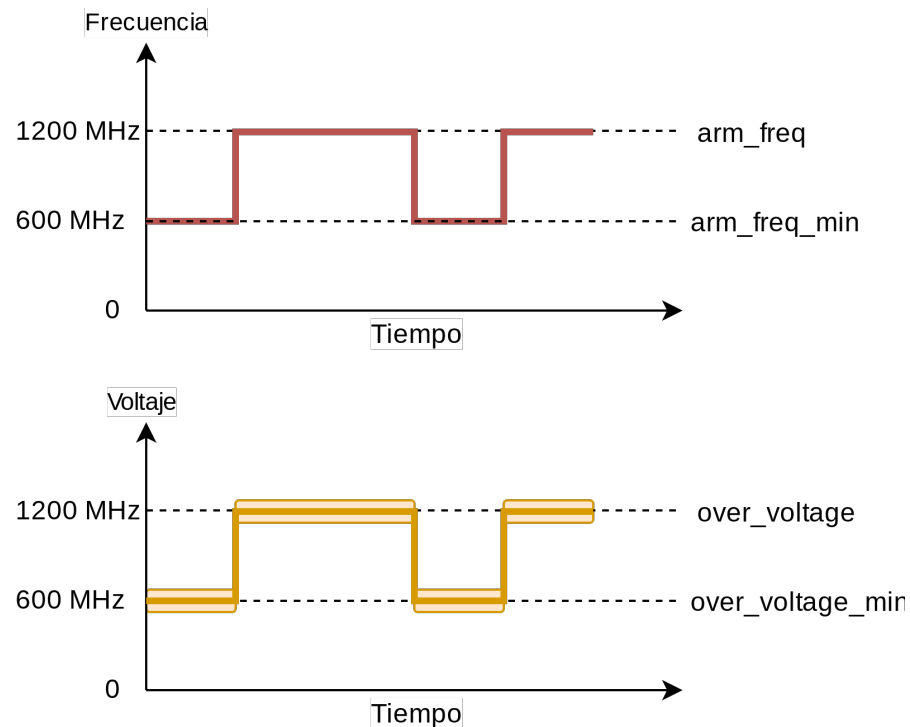


Figura 3.8: DVFS en la Raspberry Pi 3B

Las principales políticas de gestión de energía y rendimiento son *ondemand* y *performance*. En la política *ondemand* la frecuencia depende de la carga del SO mientras que en la segunda el procesador siempre funciona a la máxima frecuencia. CPUFreq también ofrece un interfaz al usuario para modificar y consultar el estado del control de potencia (ver Anexo A.3).

En el caso de la Raspberry Pi 3B las únicas frecuencias disponibles vienen determinadas por las variables `arm_freq` y `arm_freq_min` (en MHz) del fichero `/boot/config.txt`. La modificación del voltaje que alimenta al procesador (CPU+GPU) se indica mediante las variables `over_voltage` y `over_voltage_min` del mismo fichero. Sus posibles valores

están comprendidos entre -16 (0.8V) y 8 (1.4V), en incrementos de 0.025V. El valor máximo de la tensión está restringido a 1.35V, a menos que se inicialice la variable `force_turbo` a 1, pero ello conlleva la pérdida de la garantía. El fabricante tiene constancia de que se ha incumplido dicha restricción porque esa inicialización modifica un registro de forma permanente.

Cualquier cambio en el fichero `/boot/config.txt` solo tiene efecto al reiniciar el equipo.

El **SoC** tiene un sensor de temperatura, y la **GPU** ejecuta un software que encuesta el registro de este sensor para asegurarse de que la temperatura del chip no supera los 80°C. Si esto ocurriera se inicia el proceso de *thermal throttling* o ahogamiento térmico en los núcleos reduciendo tanto la frecuencia de la **CPU** como la tensión a sus niveles mínimos [19] (Fig. 3.8). En caso de superar los 85°C también se realiza *thermal throttling* en la **GPU**. La monitorización de la temperatura en los experimentos se realizan mediante el firmware `vcgencmd`.

El **DVFS** realizado en la Raspberry es muy simple ya que solo tiene dos modos de funcionamiento. Cuando el sistema está funcionando a la frecuencia máxima/mínima se aplica el máximo/mínimo voltaje (Fig. 3.8).

EXPERIMENTACIÓN

En este capítulo se detalla el equipo de medida y el software utilizado en los experimentos de la plataforma Raspberry. También se definen los experimentos a realizar, la metodología seguida para lograrlo y se analizan los resultados obtenidos.

4.1 MEDICIÓN



Figura 4.1: Analizador de potencia Newtons4th PPA520

Para la medición de la potencia se utiliza el analizador de potencia *Newtons4th PPA520* (Fig. 4.1) que permite medir dos fases simultáneamente [36]. Soporta un máximo de 20 A y 1000 V con picos de 300 A y 2500 V [37]. Consta de un pantalla para la visualización de las mediciones, pero también se puede comunicar con un ordenador mediante puerto serie, Ethernet o USB.

Las conexiones realizadas para medir la potencia consumida por la Raspberry se muestran en la Fig. 4.2. Han requerido seccionar el cable de alimentación que sale del cargador de la Raspberry (corriente continua a 5V con intensidad máxima de 2.5A), midiendo la intensidad en el polo positivo. A continuación, se colocan en paralelo los dos cables utilizados para la medición del voltaje. Las conexiones se deben hacer en este orden para evitar la pequeña caída de tensión producida en la resistencia interna utilizada por el PPA520 en la medición de la intensidad. Las mediciones realizadas con este conexionado no son muy precisas dado que se mide la potencia consumida por todo el equipo y no sólo por el procesador. En los experimentos se intenta reducir este error desactivando la conexión wifi, el entorno gráfico, la GPU y el puerto HDMI para eliminar sus respectivos consumos. Por tanto, los principales consumidores, exceptuando el procesador, serán la DRAM, el puerto RJ45 y el hub ethernet utilizado para la conexión con el ordenador personal.

Se consideró la posibilidad de cortar el raíl proveniente del regulador de tensión (componente RT8088 en la Fig. 3.7) que alimenta el

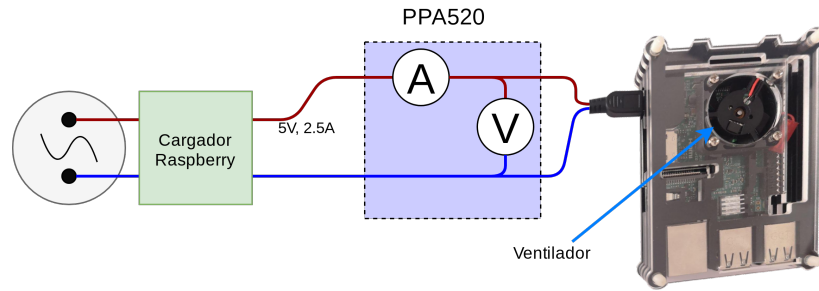


Figura 4.2: Medición de potencia en la Raspberry Pi 3B

procesador y realizar las medidas en ese punto. Esto no fue posible debido al poco espacio libre que hay en el lugar del circuito impreso donde se encuentran tales pistas.

Durante los experimentos realizados (Sec. 4.3 y 4.4) se observó que el voltaje aplicado no es el especificado en el fichero `\boot\config.txt`, sino que se realiza un ajuste de unos cuantos milivoltios según el estado térmico del SoC. Para verificar la precisión de las lecturas de voltajes realizadas vía software, se midió la tensión proporcionada al procesador por medio de los puntos de test *PP58* (positivo) y *PP4* (negativo) mostrados en la Fig. 3.7. Las diferencias entre el valor medido y el proporcionado por software son del orden de milivoltios por lo que se consideran despreciables y se valida la hipótesis de que la Raspberry Pi realiza un pequeño ajuste sobre el voltaje (*AVS*) especificado en el fichero `\boot\config.txt`.

4.2 SOFTWARE Y METODOLOGÍA

La acción de estresar o torturar un procesador consiste en ejecutar un programa que realice un uso intensivo de algunas partes del procesador para llevarlo a situaciones límites en términos de temperatura y consumo energético.

El programa utilizado para estresar la CPU es *sysbench* [34]. Este software calcula números primos y comprueba los resultados obtenidos. La ejecución de este programa provoca un calentamiento significativo del procesador, porque la parte aritmética tiene una alta actividad y apenas se interacciona con la memoria. Mediante argumentos se le puede indicar el número de primos a calcular y el número de hilos utilizados.

Dado que en los experimentos realizados en este TFG se necesita experimentar con más de dos frecuencias, siempre se trabaja con los valores máximos de frecuencia y voltaje. Para cambiar estos valores, entre experimentos se modifican los valores necesarios del fichero `\boot\config.txt` y se reinicia el sistema. Con el objetivo de agilizar los experimentos se implementaron dos *scripts* en *bash*. Uno monitoriza el voltaje, frecuencia y temperatura y el otro cambia el gobernador a *performance*, inicia el programa de estrés y almacena el registro del primer *script* en un fichero. La comunicación con la Raspberry y el lanzamiento de estos *scripts* se realizan mediante *ssh* desde un ordenador personal con *Debian 4.9*.

La recolección de las medidas proporcionadas por el analizador de potencia *PPA 520* se lleva a cabo con ayuda del programa *PPALoG* proporcionado por el fabricante. Para ello hay que especificar en la interfaz de usuario del programa el fichero destino de los datos registrados, el tiempo de monitorización, intervalo entre medidas y los datos a registrar (intensidad pico, potencia, ...). Dado que este programa es para *Windows 10*, se utiliza desde un ordenador auxiliar.

Finalmente, se implementó un *script* en *Python* para combinar los resultados obtenidos por software y los proporcionados por el aparato de medida. Este *script* también se utiliza para crear gráficas a partir de los resultados.

4.3 EXPERIMENTO 1: POTENCIA DINÁMICA Y FRECUENCIA

4.3.1 Descripción

Este experimento se hace para analizar el comportamiento de la potencia dinámica al variar la frecuencia (Ec. 2.3). Consta de cinco puntos a diferentes frecuencias en los que se toman medidas cada segundo durante tres minutos. A continuación se explica la metodología seguida en la preparación de este experimento.

Primero se elige la frecuencia mínima modificando el parámetro *arm_freq* con la nueva frecuencia. Después se reinicia el sistema para que tenga efecto. Si arranca con la nueva configuración y es capaz de ejecutar correctamente el programa *sysbench* durante 10 minutos se considera estable y los pares voltaje y frecuencia se dan por válidos. En caso de que el sistema no arranque, hay que modificar el fichero */boot/config.txt* montando la tarjeta sd en otro ordenador. En ocasiones esta tarjeta se corrompe durante un mal arranque del sistema debido a una mala configuración del voltaje y/o frecuencia. En esas situaciones, hay que instalar de nuevo el SO en la tarjeta. Estos pasos se repiten hasta encontrar la frecuencia más pequeña para la que el sistema es estable (700 MHz). Después, se repite el mismo procedimiento para encontrar la máxima frecuencia (1200). Se utiliza el voltaje por defecto (1.2V). No se puede fijar una temperatura constante para el procesador dado que el ventilador instalado en la Raspberry Pi no se puede regular.

Una vez establecido el rango de frecuencias, se procede a realizar el experimento. En primer lugar, se debe modificar el fichero */boot/config.txt* con la configuración del primer punto del experimento. Después se reinicia el sistema y se ejecuta el *script* que lanza el programa *sysbench* en cuatro núcleos. Este *script* también almacena en un fichero parámetros de monitorización como el voltaje del procesador o la temperatura. Todos estos pasos se deben realizar de forma remota mediante ssh ya que la interfaz gráfica ha sido deshabilitada (Sec. 4.1). Antes de la ejecución del *script* anterior se comienzan a tomar medidas

en el analizador de potencia utilizando el programa *PPALoG*. Este programa se ejecuta en un ordenador auxiliar con *Windows 10*. Al acabar los experimentos se consolidan todos los ficheros en uno, y se sincronizan los datos con un *script* propio implementado en *Python*. Para cada punto del experimento se toma la mediana como muestra más representativa ya que al principio del mismo hay bastante variación en la potencia medida.

4.3.2 Resultados

Las mediciones obtenidas forman claramente una recta. Aplicando una regresión lineal se obtiene $P = 0,002f + 1,737$ como ecuación de la recta. El coeficiente 0.002 (2 si f está en GHz) expresa el valor de las incógnitas ACV^2 en la ecuación:

$$P = A \cdot C_L f V^2 + P_{est} \quad (4.1)$$

El término independiente de la recta obtenida es la potencia estática que consume la Raspberry Pi con un voltaje de 1,2V.

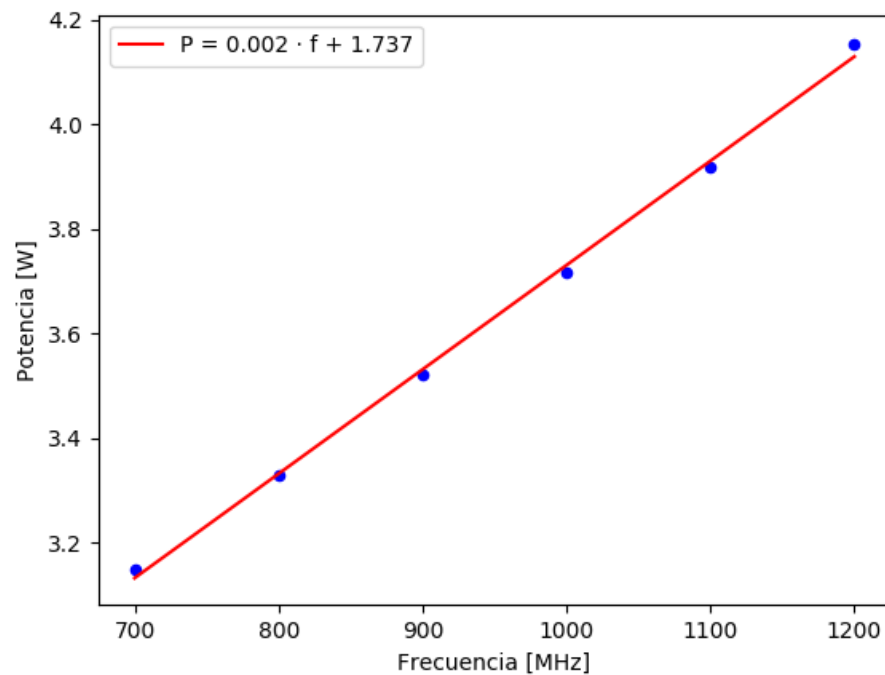


Figura 4.3: Comportamiento de la potencia al variar la frecuencia entre 700MHz y 1200MHz con tensión constante 1.2V y temperatura no controlada (varía entre 45°C y 52.5°C).

4.4 EXPERIMENTO 2: POTENCIA Y VOLTAJE

4.4.1 Descripción

El objetivo de este experimento es comprobar que la potencia dinámica aumenta cuadráticamente al incrementar el voltaje (Ec. 2.3). El experimento está compuesto por 5 puntos a diferentes voltajes. Debido a la limitación del ventilador, se utiliza la mínima frecuencia posible (700MHz) ya que con frecuencias y voltajes muy altos se calienta demasiado el procesador. Además, al establecer una frecuencia pequeña es posible usar un rango de voltajes mayor ya que el equipo es estable con los menores voltajes permitidos. En este experimento solo hay que ajustar el voltaje mínimo (1.05V). El proceso es parecido al del experimento anterior. Primero se configura el equipo con el voltaje deseado y después se realiza una prueba de estabilidad durante 10 minutos. En cada punto del experimento se toman medidas cada segundo durante tres minutos. Para cambiar la configuración durante el experimento se debe modificar el fichero `/boot/config.txt` y volver a reiniciar el equipo.

4.4.2 Resultados

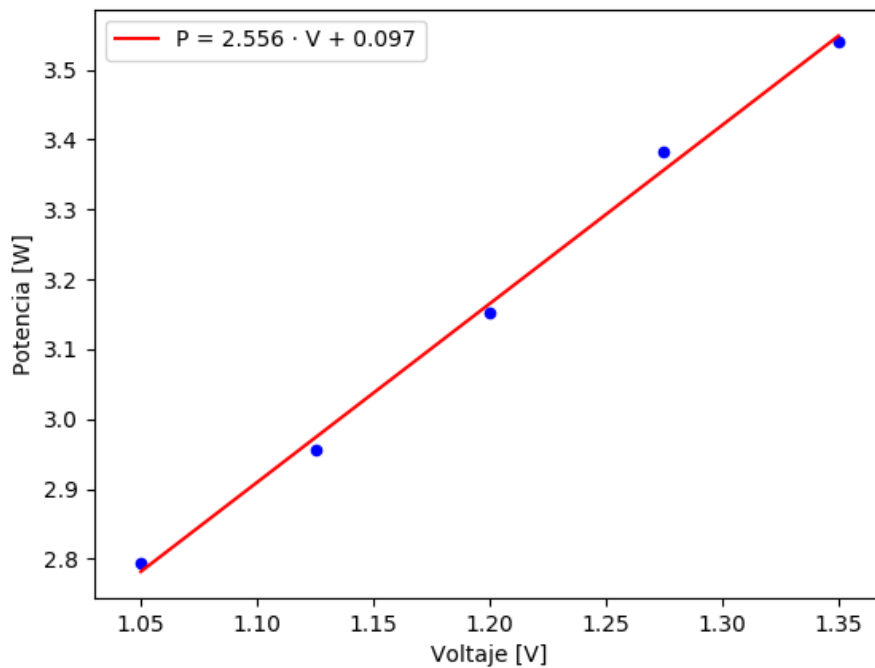


Figura 4.4: Variación de la potencia al cambiar el voltaje. Se varía el voltaje entre 1.05V y 1.35V. La frecuencia está fija a 700MHz y la temperatura no se controla (varía entre 43°C y 50°C).

Al analizar los resultados obtenidos se observa que la potencia no varía cuadráticamente con la tensión (Fig. 4.4). Esto se debe a limitaciones impuestas por la plataforma utilizada. La primera de

ellas es el pequeño rango de voltajes utilizables. Probablemente con un mayor rango de voltajes hubiera sido posible ver un crecimiento cuadrático de la potencia dinámica ya que la diferencia de consumos entre los puntos de mínimo y máximo voltaje hubiera sido mayor. La segunda limitación es la temperatura, ya que el ventilador no se puede regular y es muy pequeño.

4.5 CONCLUSIONES

El estudio y experimentación con la Raspberry Pi ha permitido un primer contacto con el problema de control de potencia y rendimiento, y la familiarización con los procedimientos de medida de potencia y variación de frecuencia en un entorno sencillo, aunque limitado. Estas limitaciones han impedido la obtención del resultado esperado en el caso del segundo experimento (Sec. 4.4).

PLATAFORMA ASUS

A continuación se describen los principales componentes del equipo profesional para *overclocking* configurado en la segunda parte del TFG. Además, se explica la gestión de energía y rendimiento que este realiza.

5.1 PLACA BASE Y COMPONENTES AUXILIARES

Se ha elegido una placa *Asus Rog Rampage VI Extreme Omega*[3]. El *chipset* soportado es *Intel x299*. Este *chipset* fue elegido porque los procesadores compatibles no incorporan *GPU*, ya que en caso contrario el calentamiento de la *GPU* podría interferir en los experimento. Se pueden instalar hasta tres tarjetas gráficas y ocho memorias *DIMM* hasta una capacidad máxima de *DRAM* de 128GB. Se han poblado cuatro zócalos con módulos de 8GB y 2400MHz para ocupar los cuatro canales[50].

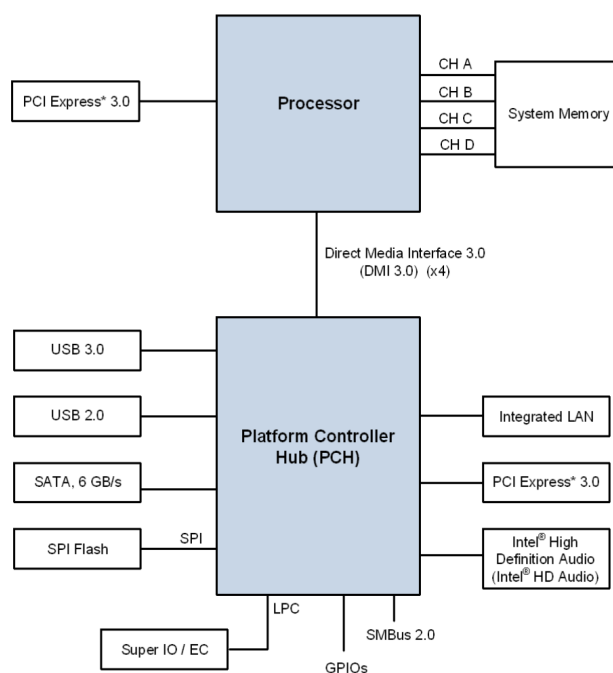


Figura 5.1: Diagrama de componentes de la Asus Rog Rampage VI Extreme Omega. Fuente: [30]

Respecto al almacenamiento, se pueden instalar hasta tres discos de estado sólido *NVMe* con formato *M.2* y seis puertos *SATA III* (6 Gb/s). Además, es posible tener un sistema de almacenamiento *RAID*

La adquisición de este equipo fue especialmente compleja. En noviembre de 2018 se realizó un primer pedido en PcComponentes que incluía la placa base Asus Rog Rampage VI Extreme. Tras más de un mes de espera, el transportista sufrió un accidente. Solicitamos la placa a riesgo de que estuviera dañada, se probó y se tuvo que devolver. Para entonces Asus había retirado esta placa del mercado debido a unos fallos con el *RAID* y la conexión de múltiples *GPUs*, y ningún proveedor la ofrecía. Asus anunció una actualización, denominada Asus Rog Rampage VI Extreme Omega, que se reservó de forma inmediata. Finalmente, a finales de enero de 2018 se recibió el equipo completo y se procedió a su montaje. Aún hubo que cambiar algunos componentes por problemas de compatibilidad.

0, 1, 5 o 10 ya que ofrece soporte hardware para ello. En este caso, se optó por un SSD de 240GB para instalar el SO y un disco duro de 1TB como almacenamiento adicional[47, 51]. En la Fig. 5.1 se muestra los componentes soportados por el *chipset*.

La principal razón para elegir esta placa es que está específicamente diseñada para realizar *overclocking* (Subsubsección 2.2.0.2), con reguladores de tensión (VRM) muy robustos. Los comentarios positivos en foros especializados en *overclocking* [42, 53] fueron refrendados por técnicos con experiencia en esta técnica a los que pudimos consultar. Además, el BIOS de Asus es uno de los mejores ya que permite realizar muchas configuraciones relacionadas con el *overclocking*. Por ejemplo, permite dibujar una gráfica para regular las revoluciones de los ventiladores en función de la temperatura de los componentes. Para refrigerar el VRM se cuenta con dos ventiladores que también se pueden regular. Además, hay una pantalla OLED en la que se muestran los códigos de error durante el arranque y la temperatura del procesador. También es posible instalar dos BIOS desde un USB con ayuda de un botón que hay en la placa.



Figura 5.2: Asus Rog Ramapge VI Extreme Omega. Fuente: [3]

Por otra parte, si el sistema se apaga debido a un problema de estabilidad durante el *overclocking*, se dispone de un pulsador para

arrancar en modo seguro estableciendo la configuración de fábrica en el BIOS. Otro pulsador permite cargar la última configuración estable del BIOS. Por último, el programa *AlSuite 3* ofrece una interfaz que permite modificar la mayoría de parámetros del BIOS desde Windows 10. Este programa también cuenta con un modo de funcionamiento en el que se realiza un *overclocking* automático.

La elección de la placa se complementó con la elección e instalación de una fuente de alimentación y de un sistema específico de disipación de calor (detalles en las Secs. A.8 y A.9).

5.2 PROCESADOR

El procesador es un *Intel Core i7-7800X* de séptima generación integrado a 14nm+. Con arquitectura *Skylake*, es la primera generación de procesadores *Intel Core Series X* para servidores, comercializada desde mitad de 2017. Combinan características de los *Intel Core* (clientes) y los *Intel Xeon* [10] (servidores) (ver Anexo. A.7).

La organización interna del procesador depende de su línea de fabricación, en función del número de núcleos: bajo (LCC), alto (HCC) y extremo (XCC). El procesador adquirido es un LCC (menos de diez núcleos). Por tanto, su distribución podría ser similar a la mostrada en la Fig. 5.3. Posiblemente este procesador contiene diez núcleos, cuatro de ellos desactivados en producción (por defecto u otros motivos). Los controladores de memoria están en ambos lados del procesador y cada uno gestiona dos canales de memoria. La conexión entre los diferentes núcleos se realiza mediante una malla 2D compuesta por anillos. En cuanto a la cache, cada núcleo tiene una L1 de datos de 32KiB y una L2 privada e inclusiva de 1MiB. Además, en cada núcleo hay una porción (1,375MiB) de la cache L3, que es compartida y no inclusiva.

5.3 DOMINIOS DE FRECUENCIA

En la placa base hay un oscilador que genera una señal de reloj fija. Esta señal atraviesa un PLL del cual se obtiene el reloj de referencia del sistema o *Base Clock* (BCLK). El BCLK es usado por el procesador, DRAM y PCH, entre otros, como señal de reloj principal (Fig. 5.4). En el PCH y DRAM hay otros PLLs que multiplican el BCLK por un valor para obtener la frecuencia de reloj adecuada. Por ejemplo, desde el PCH se suministra el reloj al DMI y PCIe. El BCLK por defecto es de 100MHz y se puede modificar con una precisión de 1MHz.

En el procesador hay un PLL más complejo que genera una señal para cada núcleo y otra para la malla de interconexión y caches de último nivel. Estas señales son múltiplos del BCLK. También se permite que todos los núcleos funcionen a la misma frecuencia.

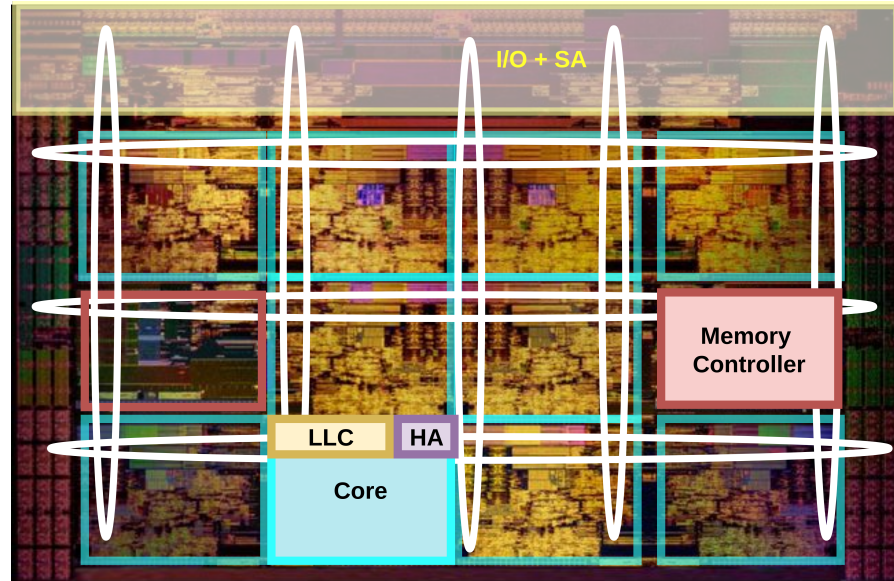


Figura 5.3: Distribución interna del Intel Core i7-7800X. Diagrama adaptado a partir del mostrado en wikichip [56]

5.4 DISTRIBUCIÓN ELÉCTRICA

La mayoría de componentes del equipo son alimentados por el cable ATX de 24 pines que conecta la placa base y la fuente de alimentación. La PSU comienza a suministrar corriente cuando el pin *PS_ON* se conecta al pin al negativo, es decir, al pulsar el botón de encendido de la placa. Este cable está compuesto por tres raíles diferentes: 3.3V, 5V y 12V. Con el raíl de 3.3V se alimentan dispositivos de pequeño consumo como los módulos DIMM o tarjetas PCIe. El de 5V se utiliza principalmente para discos duros. Por último, el raíl de 12V alimenta ventiladores de gran tamaño y el PCIe usado por la GPU. La corriente suministrada por estos raíles atraviesa unos reguladores de tensión conmutados, que reducen la tensión de entrada a la requerida por los componentes que lo van a utilizar. Por ejemplo, en la placa base analizada hay un regulador para los dispositivos de audio, otro para el generador de señales de reloj global y otro para los módulos de DRAM.

El procesador se alimenta mediante un cable EPS12V de 8 pines de los cuales cuatro son positivos y otros cuatro negativos, con una diferencia de potencial entre ellos de 12V.

Al encender el equipo, transcurre un pequeño período transitorio durante el cuál la tensión suministrada no es estable. Una vez finalizado este período, la placa base pone en alto la señal del pin *POWER_GOOD* del cable de 24 pines y se comienza el inicio del sistema. Si esta señal no está en estado alto, se fuerza el reinicio del sistema, que no tiene lugar hasta que dicha señal alcanza el valor alto. La corriente suministrada al procesador atraviesa el módulo de regu-

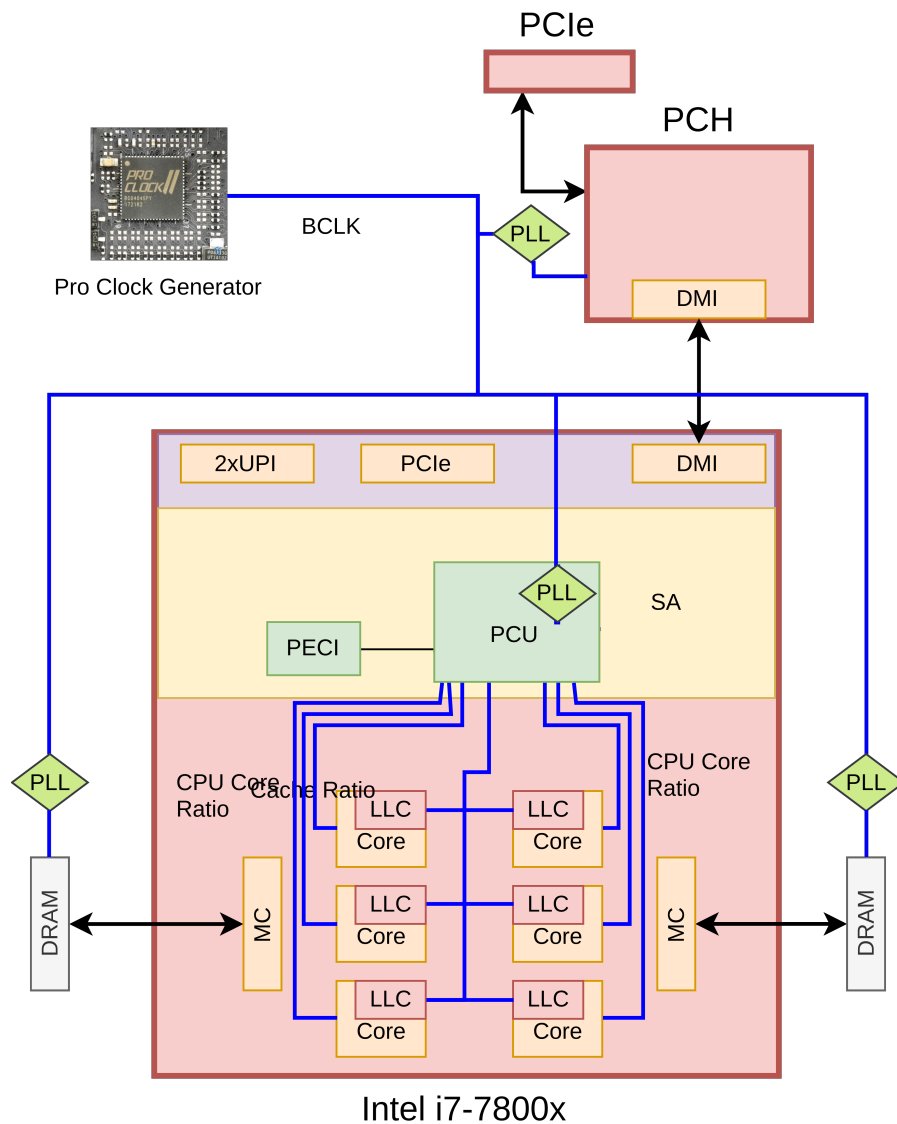


Figura 5.4: Árbol con la distribución de las señales de reloj en la Asus Rampage VI Extreme Omega. Fuente de la imagen del generador de reloj: [6]

lación de voltaje (**VRM**), del que salen cinco raíles. Los raíles *VCCD1* y *VCCD2* alimentan los dos controladores de memoria que hay en el procesador. Los raíles *VCCIO* y *VCCSA* suministran la corriente a las zonas de entradas-salidas y a la zona del sistema agente del procesador. Por último, el raíl *VCCIN* alimenta principalmente a los núcleos, caches de último nivel y red de interconexión del procesador. Las regulaciones de voltaje realizadas por el **VRM** son controladas por un microcontrolador externo llamado **TPU**. Este microcontrolador ajusta la tensión final de cada regulador del **VRM** y otros parámetros como la frecuencia de conmutación de las fases o la calibración de la línea de carga (Sec. 5.4.2). La parte del **VRM** que se encarga de la regulación del *VCCIN*, compuesto por 16 fases conectadas en paralelo que

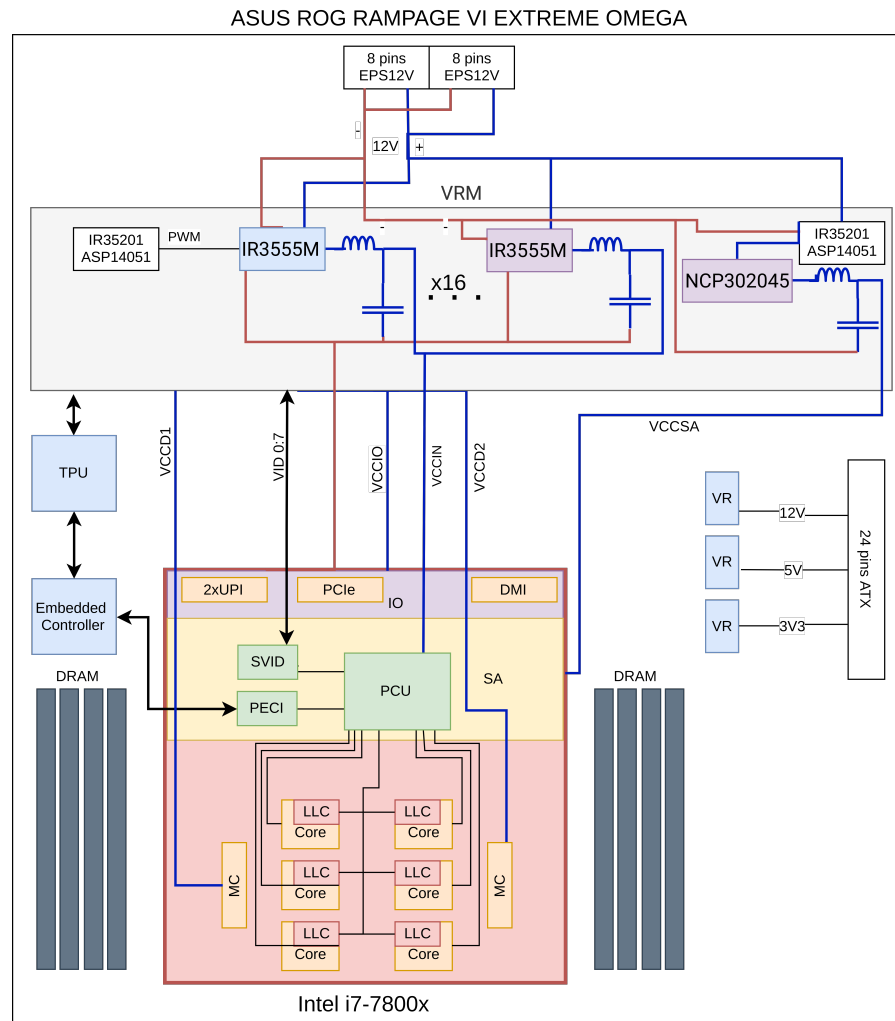


Figura 5.5: Distribución eléctrica en la Asus Rog Rampage VI Extreme Omega.

transforman de 12V a 1.85V aproximadamente. El microcontrolador *IR35201* o *DIGI+ Power Control* genera 16 señales *PWM* diferentes con una frecuencia máxima de 1MHz en función de las órdenes recibidas del *TPU*. Cada una de estas señales controla una de las fases, que trabajan de forma secuencial. Es decir, en un instante dado solo transforma una de las fases mientras los transistores de potencia (*IR3555*) del resto de las fases se están enfriando [46]. Por tanto, cuantas más fases tenga un *VRM* es más probable conseguir una señal estable ya que los transistores se calientan menos. También incrementa la corriente máxima suministrable.

En el procesador el raíl *VCCIN* se divide en varios dominios. El encargado de la gestión de energía dentro del procesador es la Unidad de Control del Paquete o *PCU*. Este reside en el dominio del sistema agente y es un microcontrolador integrado con un hardware específico para ejecutar unas máquinas de estados. Está conectado con todos los núcleos y otros bloques funcionales a través de los Agentes de



Figura 5.6: Módulo de regulación del voltaje en Asus Rog Rampage VI Extreme Omega. Fuente: [6]

Gestión de Energía (**PMAs**). Los **PMAs** recolectan información como el consumo de energía y temperatura y ejecutan funciones de control. Estas **PMAs** contienen puertas para apagar los núcleos individualmente o porciones de la cache de último nivel (L3). La **PCU** también se comunica con los VRM y el BIOS de la placa base. La **PCU** solicita al VRM el voltaje requerido por cada uno de los raíles usando el protocolo **SVID** (Sec. 5.4.1). Con esta gestión de energía se consigue una mayor eficiencia ya que un núcleo puede estar ejecutándose a un voltaje alto mientras que el resto pueden estar apagados o en un estado de reposo debido a la poca actividad del procesador (Sec. 5.5):

Mediante los contadores **RAPL**, Intel recolecta eventos hardware y los combina junto con el voltaje, frecuencia y temperatura para obtener una aproximación de la potencia consumida. Esta información se utiliza para decidir si en un núcleo determinado se va a ejecutar a la frecuencia Turbo (Sec. 5.5) o si hay que iniciar el proceso de ahogamiento térmico (Sec. 5.5).

5.4.1 Serial Voltage Identification (SVID)

SVID es un protocolo de comunicación serie y bidirrecional utilizado por el procesador para especificar al VRM el voltaje que este le debe suministrar. Está compuesto por la señal de reloj, el bus de datos de 8 bits y una señal de alerta (*clock, data, alert*). A través del bus de datos, la PCU pide a los reguladores externos el voltaje más adecuado basándose en condiciones como la temperatura o frecuencia del procesador. Después de cada petición del procesador por el bus de datos, el VRM debe enviar de vuelta un mensaje de confirmación. El identificador de asociado a cada voltaje está almacenado en una tabla tanto en el procesador (*fuse unit*) como en la placa base (Fig. 5.7).

Las especificaciones del SVID descritas pertenecen a la versión VR12.0 porque no se ha encontrado documentación oficial de la versión utilizada por el procesador instalado (VR13.0). Los cambios presentes en la nueva versión probablemente sean solo características técnicas como el mínimo incremento o decremento de voltaje o el tiempo de respuesta de los VRM.

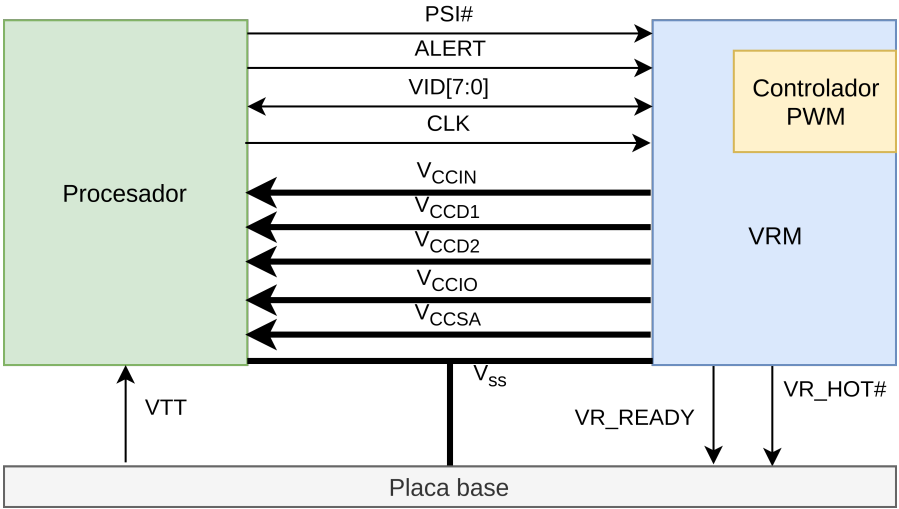


Figura 5.7: Conexiones del protocolo de comunicación SVID.

Con este protocolo se pueden gestionar hasta 16 raíles de alimentación de forma simultánea. Aunque, antes de enviar información relacionada a otro raíl por el bus de datos, el procesador debe informar al VRM del cambio de raíl. En la tabla 5.1 se muestran los valores máximos y mínimos permitidos para los raíles del procesador utilizado en este TFG. La especificación del protocolo SVID [54] permite solicitar voltajes hasta 3.04V aunque estos no son utilizados en este equipo.

	Mín	Máx	Unidad
VCCIN	-0.3	2.15	V
VCCD	-0.3	1.35	V
VCCIO	-0.3	1.35	V
VCCSA	-0.3	1.35	V

Cuadro 5.1: Tensiones máximas y mínimas permitidas

	Mín	Máx	Unidad
VCCIN	1.55	1.80	V
VCCD	1.05	1.20	V
VCCIO	0.95	1.00	V
VCCSA	–	–	V

Cuadro 5.2: Rango de voltajes de funcionamiento de los raíles de alimentación familia Skylake-X

En la tabla 5.2 se muestran el rango de voltajes utilizado por cada raíl en la familia Skylake-X.

Además, este protocolo tiene cuatro modos de funcionamiento: *PS0*, *PS1*, *PS2* y *PS4*. *PS0* es el estado de máximo rendimiento y la *VRM* debe ser capaz de responder a incrementos o decrementos en la tensión de $25\text{mV}/\mu\text{s}$ para el raíl V_{CCIN} y de $10\text{mV}/\mu\text{s}$ para el resto de raíles. Para notificar al *VRM* del modo de funcionamiento se utiliza la señal *PSI#* y mediante la señal *PROCHOT#* el procesador notifica al *VRM* de una situación de sobrecalentamiento para que este inicie también el proceso ahogamiento térmico. La señal *VR_HOT#* es similar a la *PROCHOT#* pero ahora el dispositivo sobrecalentado es el *VRM*.

Ante un cambio brusco del nivel de carga de un procesador, se produce un incremento (decremento) en la corriente y un decremento (incremento) de la tensión solicitada a los *VRM*. Para suavizar estos picos de tensión e intensidad, el *TPU* monitoriza la tensión de salida y en caso de detectar un cambio ajusta de nuevo los parámetros del *VRM* para estabilizar de nuevo la tensión. Estos picos disminuyen al incrementar la frecuencia de trabajo del *VRM* ya que estas variaciones de la tensión nominal se detectan y regulan más rápido.

Para evitar alcanzar el valor mínimo o máximo permitido por el procesador cuando se dan estas situaciones, el controlador del *VRM* le aplica un offset (V_{offset}) a la tensión proporcionada al procesador. De esta forma, cuando el procesador solicita al *VRM* una tensión próxima al valor máximo soportado, realmente se le está suministrando un voltaje menor mientras que al solicitar un voltaje muy bajo se le aplica un offset positivo (Fig. 5.8).

Este control que realiza el *VRM* es muy importante en los equipos usados para *overclocking* o *undervolting* ya que se somete al procesador a una gran carga de trabajo y los voltajes utilizados son cercanos a los valores límites.

Además, hay que tener en cuenta que los raíles que alimentan al procesador tienen una pequeña resistencia (R_{LL}) que provocan que la tensión solicitada por el procesador sufra una caída de tensión (V_{droop}) que es proporcional a la carga del sistema (Ec. 5.1).

$$V_{cc} = VID - V_{droop} = VID - I_{CC}R_{LL} \quad (5.1)$$

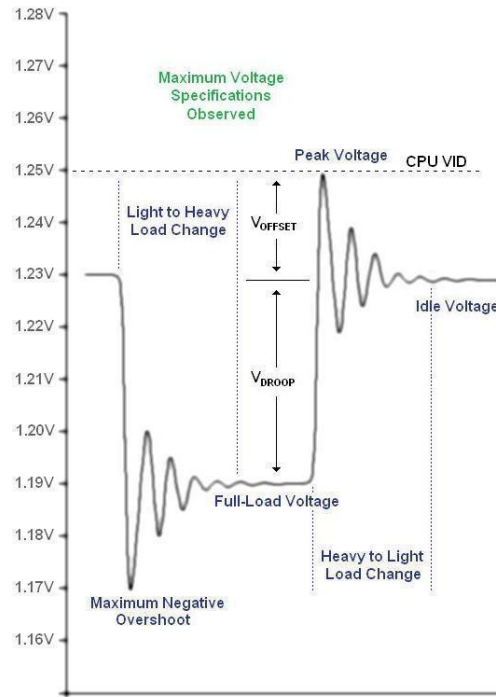


Figura 5.8: Transición voltaje de plena carga a reposo con el protocolo SVID.
Fuente: [7]

La regulación de esta caída de tensión se realiza mediante la calibración de la línea de carga (Sec. 5.4.2).

5.4.2 Calibración de la línea de carga

La calibración de la línea de carga o *Load-Line Calibration (LLC)* es un mecanismo usado por el controlador del VRM para contrarrestar la caída de tensión que hay entre este y el procesador. Para ello, el VRM en lugar de proporcionar el voltaje especificado por el SVID le aplica un offset al voltaje para que al procesador le llegue el voltaje especificado mediante el interfaz SVID.

Desde el BIOS se puede establecer la máxima variación permitida entre el voltaje pedido y recibido. Reducir mucho esta diferencia de potencial puede ser peligroso para el equipo dado que al operar con voltajes muy altos/bajos siguen existiendo picos de tensión debidos a las variaciones de carga del sistema, que pueden alcanzar valores no permitidos por procesador (Fig. 5.9).

5.5 CONTROL DE POTENCIA Y RENDIMIENTO

El control de potencia y rendimiento en la placa Asus tiene tres modos de funcionamiento en función del agente que toma las decisiones del control (SO, PCU o BIOS).

Los estados para el control de potencia y rendimiento definidos por Intel pueden variar de los definidos en el estándar ACPI

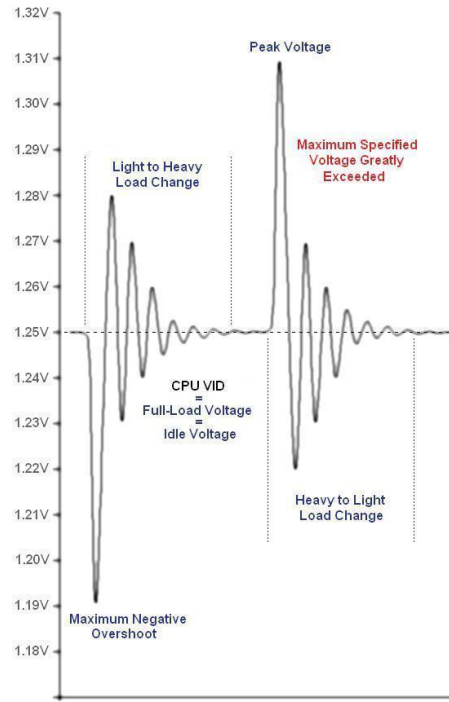


Figura 5.9: Transición voltaje de plena carga a reposo usando LLC.
Fuente: [7]

En el modo por defecto, el SO realiza el control de potencia y rendimiento. Para ello, se utiliza la tecnología *Intel Speed Step* (Sec. 5.5.2), que es la implementación del estándar *ACPI*. Con este modo de funcionamiento se utilizan los estados *ACPI* en los que cada núcleo opera a una frecuencia y voltaje establecidos por fabricante (Anexo A.4). El cambio a un nuevo estado lo decide el subsistema del SO encargado del control de potencia y rendimiento. Para ello, se requiere un controlador que aplique los cambios necesarios en el hardware.

Los estados de potencia y rendimiento de este procesador son *PC0*, *PC1*, *PC3* y *PC6*. En el estado *PC0* al menos un núcleo está ejecutando instrucciones. En cambio, el *PC6* es el estado de reposo más profundo en el que ciertas partes del procesador están apagadas. También hay unos estados a nivel de núcleo: *CC0*, *CC1*, *CC3* y *CC6*. Un estado PC o CC mayor implica mayor ahorro energético y mayor latencia de transición al estado de funcionamiento. Pero, volver al estado *CC0* desde un estado CC profundo cuesta unas décimas de microsegundos durante las cuales el núcleo no está ejecutando instrucciones. Además, esta transición entre estados conlleva un mayor consumo energético. Por tanto, una transición a un estado CC profundo para un corto período de tiempo, puede resultar en un mayor consumo de energía que la ahorrada durante la estancia en dicho estado.

Intel cuenta con un algoritmo que calcula la energía consumida por una transición, en base al cual puede ignorarse una orden del

sistema operativo si el PCU considera que es ineficiente en términos energéticos. El estado del procesador se define en función del estado de menor ahorro energético de todos los núcleos. Es decir, si un núcleo está en CC₁ y resto en CC₃ el estado del procesador será PC₁.

El segundo modo utiliza la tecnología *Intel Speed Shift* (Sec. 5.5.3). En este caso, se le especifica al PCU un rango de rendimiento y un valor de eficiencia energética. Después, este realiza el control necesario para que se cumplan dichas condiciones. Estos valores pueden ser modificados por el SO mediante registros de modelo específico o MSR. Esta técnica no supone una gran mejora en cuanto al ahorro energético global [23]. Sin embargo, en sistemas con muchos picos de cargas se consigue una mejora significativa en el rendimiento ya que los cambios de frecuencias tienen menos latencia. Esto se debe a que el ajuste lo realiza un hardware dedicado y se evita la traza de llamadas al sistema del método anterior.

En el último modo de funcionamiento, el BIOS indica al PCU la frecuencia y el voltaje. Esto se realiza mediante el bus PECI que conecta el procesador con el controlador empotrado de la placa base. En este modo, cada núcleo tiene un estado de reposo y otro de pleno rendimiento [5]. La frecuencia máxima de cada núcleo se ajusta multiplicando la señal BCLK por un entero. El ajuste del voltaje desde el BIOS se puede realizar de tres formas diferentes:

- Manual: el usuario especifica un voltaje para cada núcleo.
- Offset: el usuario especifica un offset que se le aplica al voltaje nominal.
- Adaptativo: el PCU regula el voltaje de forma automática. Este modo de funcionamiento es parecido al utilizado por la tecnología Intel Speed Shift.

El control de temperatura lo realiza el PCU junto con el microcontrolador DTS.

5.5.1 Sensor Térmico Digital

El Sensor Térmico Digital o DTS es un microcontrolador integrado en el procesador que monitoriza varios sensores de temperatura con una resolución de 1°C [12]. Cuando alguno de estos alcanza una temperatura crítica se activa el Circuito de Control Térmico o TCC para reducir la temperatura. Para ello hay dos mecanismos diferentes: *Thermal Monitor 1 y 2* (TM₁ y TM₂). El primero utiliza la técnica de *clock gating* y el segundo hace DVFS (Sec. 2.2). Las temperaturas umbrales a partir de las que se activan los mecanismos TM₁ y TM₂ se modifican en el MSR IA32_THERM_INTERRUPT. Además, Intel usa el interfaz

de comunicación [PECI](#) para que componentes de la placa base como los ventiladores monitoricen la temperatura del procesador y regulen su velocidad.

Mediante la señal *PROCHOT#* se le indica al [VRM](#) que disminuya la potencia suministrada ya que el procesador se ha sobrecalentado. Con temperaturas muy cercanas a la *T_{JUNCTION}*, se envía la señal *THERMALTRIP_N* a la placa base y el equipo se apaga inmediatamente.

T_{JUNCTION} es la temperatura máxima a la que un circuito electrónico puede operar sin que sufra daños irreparables.

5.5.2 Enhanced Intel SpeedStep Technology

Enhanced Intel SpeedStep Technology ([EIST](#)) es la implementación de Intel de los estados P o *performance* de [ACPI](#) [[13](#), [16](#)]. Para activar esta tecnología, el [BIOS](#) durante el arranque debe escribir un 1 en el bit 16 del [MSR IA32_MISC_ENABLE](#). A continuación, el procesador comienza a funcionar en el estado establecido por defecto. Cada vez que el subsistema del [SO](#) encargado del control de potencia y rendimiento quiera realizar un cambio de estado debe escribir en el [MSR IA32_PERF_CTL](#) el identificador del estado objetivo. El estado actual se consulta leyendo el [MSR IA32_PERF_STATUS](#).

Los puntos de operación permitidos (pares frecuencia y voltaje) están almacenados en el [BIOS](#). Además, en el procesador hay un [MSR](#) con los puntos de máximo y mínimo de rendimiento para asegurar que el [BIOS](#) no permite establecer frecuencias y voltajes no permitidos.

La regulación del voltaje y frecuencia se realiza de manera independiente [[22](#)] para reducir el tiempo de indisponibilidad del procesador durante la transición de un estado a otro (unos 10 μ s). Esta reducción se debe a que el [PCU](#) incrementa el voltaje antes que la frecuencia. Por tanto, el procesador puede seguir trabajando inmediatamente después de cambiar la frecuencia ya que se tiene un voltaje suficientemente alto para que el sistema no tenga problemas de estabilidad (Fig. 5.10).

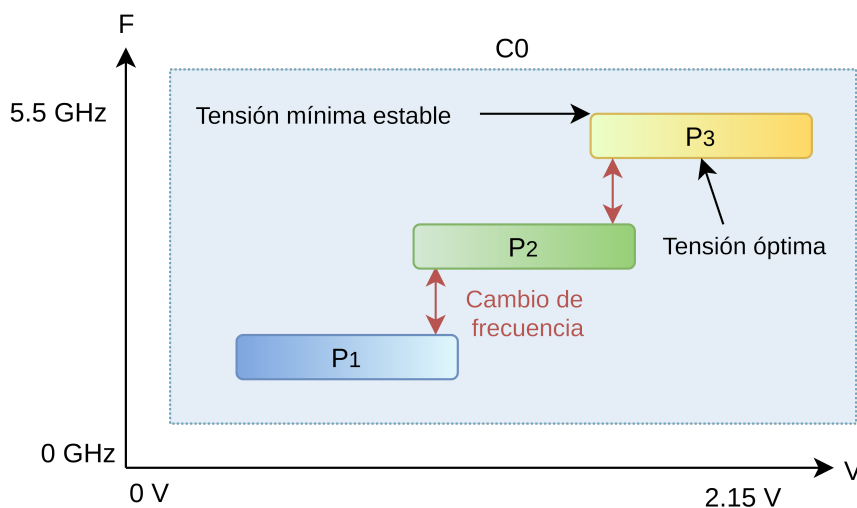


Figura 5.10: Cambio de frecuencia usando [EIST](#)

La utilización de esta técnica para el control de potencia y rendimiento supone una mejora en el rendimiento y ahorro energético debido a que en momentos de mucha carga los núcleos que sean necesarios operan en estados de alto rendimiento y después vuelven a estados de menor rendimiento o hibernación.

5.5.3 Intel Speed Shift Technology

En los manuales de Intel, esta tecnología es nombrada como de estados de rendimiento controlados por hardware o Hardware-controlled performance states. Intel Speed Shift Technology no es compatible con el estándar ACPI.

Intel Speed Shift Technology o [ESST](#) es una alternativa o mejora de la [EIST](#) (Sec. 5.5.2). Este nuevo mecanismo de control de potencia y rendimiento no tiene unos estados fijos sino que opera en un rango de frecuencias y voltaje controlados de forma autónoma por el hardware específico del procesador (PCU) [[13](#), [23](#)].

A continuación, se detalla el funcionamiento básico del controlador que el [SO](#) necesita para utilizar [ESST](#). El sistema lee el [MSR](#) `IA32_HWP_CAPABILITIES` para averiguar el rango de rendimiento del procesador. Este registro contiene 4 campos: *Guaranteed_Performance*, *Most_Efficient_Performance*, *Highest_Performance* y *Lowest_Performance*. Los últimos dos especifican el rango de rendimiento que se le puede indicar al procesador y el primero hace referencia al rendimiento mínimo garantizado. Para cambiar el rango de rendimiento, el controlador debe modificar el [MSR](#) `IA32_HWP_REQUEST` de cada núcleo. Este registro contiene, entre otros, los siguientes campos: *Minimum_Performance*, *Maximum_Performance*, *Desired_Performance* y *Energy_Performance_Preference*. Estos campos son parecidos a los anteriores. *Energy_Performance_Preference* indica la política con la que se realiza un cambio de frecuencia. Cuando su valor es bajo se prioriza el rendimiento. Esto conlleva que ante un pico de carga se hará un cambio brusco de frecuencia y voltaje. En cambio, cuando este valor sea el máximo (0xFF) los cambios de estados serán más conservadores.

El campo *Desired_Performance* indica el nivel de rendimiento ideal para cada núcleo controlado. El control aplicado por el [PCU](#) es una fusión de los dos requisitos anteriores y otros parámetros externos como la temperatura y el nivel de potencia.

Esta tecnología tiene prioridad sobre la [EIST](#) cuando ambas son activadas desde el [BIOS](#). Su uso no supone en general un gran ahorro energético en el consumo total del sistema. En cambio, en equipos con muchos picos de carga sí que se consigue una mejora considerable debido a que la transición de la frecuencia mínima a la máxima se baja de 100ms con [EIST](#) a 35ms con [ESST](#) [[23](#)]. Esta reducción se debe a que ya no hay una traza de llamadas del [SO](#) para cambiar de estado dado que el hardware realiza todo este control.

5.5.4 Intel Turbo Boost Max Technology 3.0

Es una característica de casi todos los procesadores de Intel que permite que al menos uno de los núcleos incremente su frecuencia de manera automática durante un tiempo si se cumplen ciertas condiciones de potencia, temperatura y corriente consumida por el procesador.

El periodo durante el cual los núcleos elegidos permanecen en un estado de rendimiento superior viene marcado por tres parámetros configurables desde el BIOS: los niveles de potencia y temperatura y el tiempo de la ventana de alto rendimiento.

En los procesadores Intel, la potencia máxima consumida viene determinada por dos niveles de potencia. El nivel I coincide con el TDP, es decir, la potencia consumida por el procesador en condiciones normales será inferior a esta. En cuanto al nivel II, el fabricante de la placa base lo ha establecido en $1,25 * TDP$, es decir, 175W. Este nivel indica la potencia máxima que el procesador puede consumir durante el periodo de tiempo establecido por la ventana de alto rendimiento. Algo parecido ocurre con los niveles de temperatura (Sec. 5.5.1).

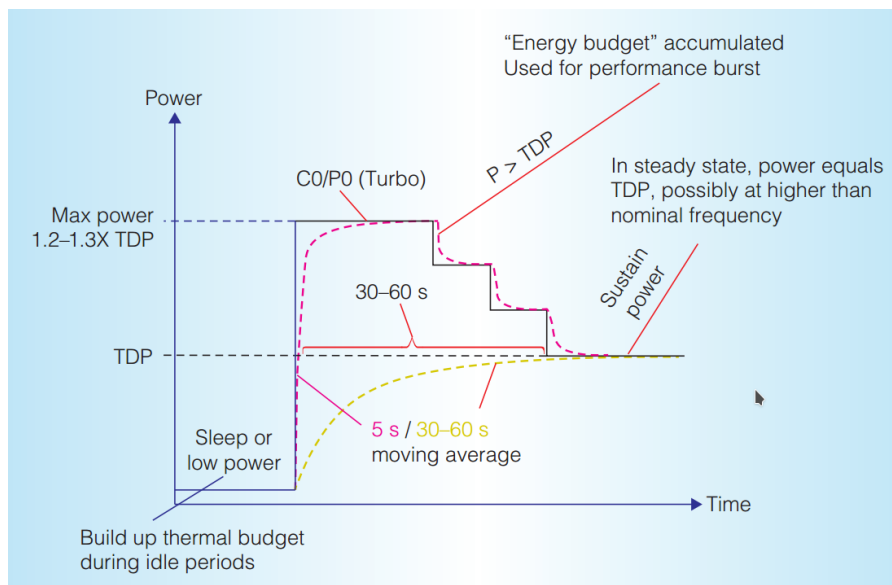


Figura 5.11: Comportamiento dinámico del Intel Turbo Max. 3.0

En caso de que el nivel térmico o el de potencia sean superados durante un periodo de tiempo mayor al establecido por la ventana de alto rendimiento se procede a activar los circuitos de ahogamiento térmico o de potencia.

La tecnología Turbo Boost se activa solo cuando el sistema se encuentra en el nivel I y la utilización del procesador es superior a cierto umbral (Fig. 5.11).

En resumen, desactivar las tecnologías Intel Turbo y SpeedShift durante los experimentos es crucial ya que permite establecer una frecuencia fija en los núcleos del procesador.

EXPERIMENTACIÓN Y RESULTADOS

En este capítulo se explica la instrumentación necesaria para medir potencia en la plataforma Asus. Después, se exponen los programas utilizados en los experimentos. Al final del capítulo, se definen los experimentos a realizar y se analizan los resultados obtenidos.

6.1 MEDICIÓN

La placa base se conecta a la fuente de alimentación utilizando entre dos y tres mangueras. La manguera principal presenta 24 pines y se usa para suministrar corriente a todos los componentes excepto al procesador. Para alimentar al procesador se utiliza la manguera *EPS12V* de 8 pines, de los cuales cuatro son positivos (12V) y cuatro negativos. Antes de llegar al procesador, esta corriente atraviesa los reguladores de tensión de la placa base para ser transformados de 12V a 1.85V aproximadamente (Sec. 5.4). Al principio se utilizó solo una de las dos mangueras para alimentar al procesador. La experimentación llevó a añadir la manguera *EPS12V* auxiliar, ya que al sobrepasar los 140W de consumo del procesador la tensión de entrada al *VRM* era inestable y llegaba a bajar hasta los 10,5V.

Para la medición de potencia se seccionaron las dos mangueras *EPS12V* para conectarlas al analizador de potencia *Newtons 4th PPA 520* (Fig. 6.1). Dado que cada manguera *EPS12V* está compuesta por ocho cables, se instaló una borna repartidora con dos polos (Fig. 6.2) y 10 conexiones por polo [15]. Cada uno de los polos (barra con 10 orificios) se utiliza para conectar los cuatro cables de cada manguera *EPS12V*. Por tanto, en el polo superior de la borna se juntan los ocho cables positivos de alimentación en el mismo punto eléctrico y en el polo inferior se reparten de nuevo para alimentar al procesador por los conectores *EPS12V* de la placa base. Entre ambos polos se conecta el amperímetro. Medir el voltaje requiere conectar el polo inferior de la borna con uno de los cuatro cables negativos de alguna de las mangueras *EPS12V*.

Con este conexionado se mide la potencia consumida por el procesador y el *VRM*. Para medir solo la potencia consumida por el procesador se intentó analizar el circuito impreso de la placa base con el objetivo de desoldar algún componente entre el *VRM* y medir en ese punto. Debido a la compleja circuitería y reducido espacio no se encontró ningún punto en el que se pueda desoldar y conectar el analizador de potencia.

En la figura 6.3 se muestra el montaje final de la plataforma Asus.

Inicialmente se consideró soldar los cables, pero al estimar que la intensidad conjunta de las dos mangueras podría alcanzar los 12A se optó por la borna repartidora. No sólo evita calentamientos en los puntos de la soldadura sino que permite conectar y desconectar fácilmente las mangueras EPS12V.

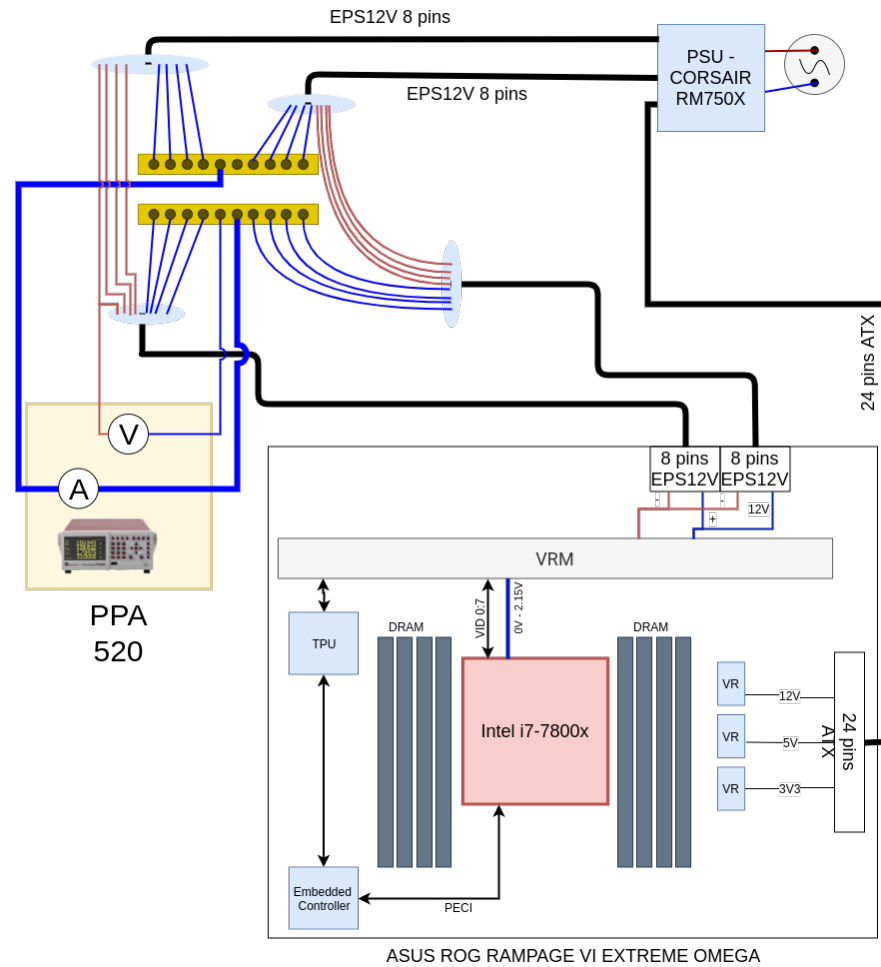


Figura 6.1: Conexionado eléctrico para la medición de potencia en la placa Asus.

6.2 SOFTWARE

En esta sección se va a exponer el software empleado en los experimentos. Primero se trata el programa utilizado para *torturar* el procesador y después se explica el proceso de recolección de datos seguido en los experimentos. La mayoría de los programas para ejecutar pruebas y estresar un computador suelen ser estar disponibles solo para *Windows* y *macOS*, lo que determinó el SO utilizado (*Windows 10*).

Uno de los programas utilizados es *AI Suite 3* [4], un software propietario de Asus que proporciona una interfaz del BIOS desde *Windows 10*. Permite modificar parámetros similares a los nativos del BIOS (Sec. A.6). *AI Suite 3* presenta cuatro secciones: monitorización de sensores, ajustes del VRM, regulación de los ventiladores y zona para *overclocking*. En este TFG se utiliza la sección de los ventiladores, que permite dibujar una gráfica para especificar la velocidad del ventilador en función de la temperatura. También se utiliza la ventana de *overclocking* para modificar el voltaje y frecuencia de cada núcleo. Además, hay una opción



Figura 6.2: Conexión de las mangueras EPS_{12V} en la borna repartidora. En la parte superior está la entrada al amperímetro y en la inferior la salida.

para establecer una configuración óptima de todos los parámetros y obtener la mejor relación entre consumo y rendimiento.

6.2.1 Tortura del procesador

Para tomar la mejor decisión se hizo un pequeño análisis parecido al realizado en [55]. Las medidas y análisis de potencia consumida por el procesador al ejecutar los principales programa de estrés disponibles se encuentran en el A.5 llevó a utilizar *Prime95* porque era el programa que más potencia media consumía. Además, a diferencia de los otros programas, el consumo era constante durante toda la ejecución.

Prime95 busca números primos de Mersenne y al encontrar uno utiliza la *prueba de Lucas-Lehmer* para verificar su validez [57]. No sólo estresa el procesador sino que comprueba la estabilidad del mismo, ya que ante un falso positivo o negativo muestra un mensaje en el informe generado. Puede ser ejecutado por línea de comandos o mediante interfaz gráfica. Los parámetros configurables son el número máximo de primos a buscar, el número de hilos o búsquedas de primos a ejecutar y el tamaño de las estructuras de datos utilizadas. Para este TFG el tamaño de los datos a utilizar se ha configurado para que quepan en la cache L2. También se habilita el uso de instrucciones pertenecientes a la extensión vectorial AVX_{512} . De esta forma, se ejecutarán muchas instrucciones de cálculo, pocos saltos y apenas operaciones

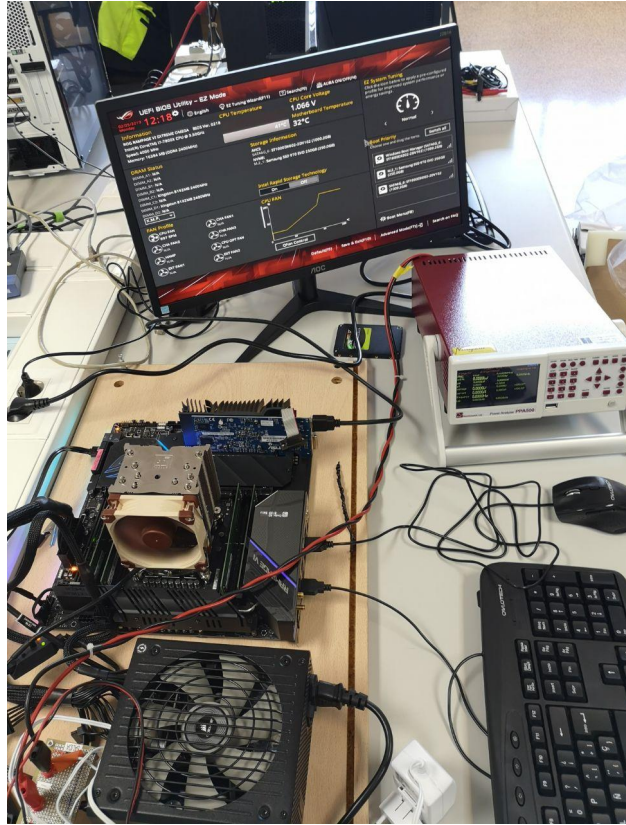


Figura 6.3: Entorno experimental de la plataforma Asus

de memoria, consiguiendo así un alto consumo de los núcleos del procesador.

6.2.2 Recolección de datos

Los datos registrados en los experimentos provienen del analizador de potencia y de los contadores de eventos de Intel. Por una parte, se mide la potencia con el *PPA 520* y se leen en tiempo real con ayuda del programa *PPALoG*, en un proceso idéntico al descrito en la Sec. 4.1. El resultado de esta medición es un fichero con el tiempo, voltaje, intensidad y potencia registradas por el *PPALoG* cada segundo. Por otra parte, también se toman medidas de sensores y eventos hardware utilizando los siguientes programas *HWiNFO* e *Intel VTune Amplifier*.

HWiNFO es un programa con una interfaz gráfica que permite ver en tiempo real el valor de la mayoría de sensores del equipo [25]. Muestra la temperatura y voltaje de cada núcleo y permite almacenar los valores en un fichero CSV. Se probaron otras alternativas como *CPU-Z* pero *HWiNFO* fue la más completa.

Intel VTune Amplifier es un programa desarrollado por Intel que realiza un análisis de rendimiento basado en los eventos hardware e identifica los cuellos de botella en el procesador. Una de las opciones

más útiles es la de ejecutar un programa y registrar información sobre él. Por ejemplo, en este proyecto se utiliza para ejecutar *Prime95* y registrar durante un tiempo determinado métricas como el número de instrucciones ejecutadas o la ocupación del procesador.

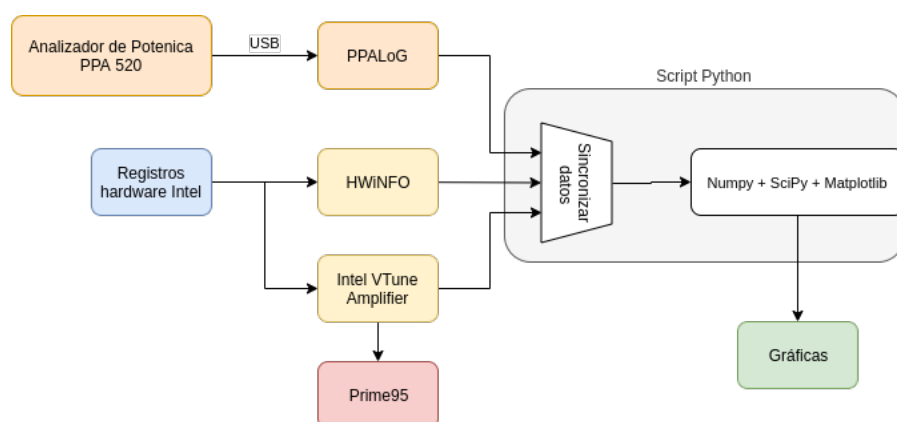


Figura 6.4: Flujo de datos en los experimentos con la plataforma Asus.

Para analizar los datos experimentales, se implementó un *script* en *Python* que lee los ficheros CSV almacenados por cada programa y sincroniza los datos. La sincronización no se puede hacer utilizando el tiempo registrado en cada fichero ya que el reloj del analizador de potencia cambia su valor durante algunos experimentos. Por tanto, se implementó un algoritmo para sincronizar las medidas basándose en el brusco cambio de la potencia consumida por el procesador cuando este empieza a ejecutar el programa de tortura. Finalmente, con los datos del experimento ya filtrados se genera la gráfica correspondiente. La Fig. 6.4 muestra el flujo de datos durante los experimentos.

6.3 EXPERIMENTO 1: POTENCIA DINÁMICA Y FRECUENCIA

6.3.1 Descripción

Con este experimento se pretende obtener una caracterización de la potencia dinámica al variar la frecuencia. Para ello, hay que variar la frecuencia del procesador manteniendo la temperatura constante, para que no varíe también la potencia estática (dependiente de la temperatura) y enmascare los resultados.

En cada fase del experimento se varía la frecuencia desde 1,2GHz hasta 3,2GHz en incrementos de 200MHz. La temperatura se mantiene constante a 60°C mediante la funcionalidad de regulación de los ventiladores del programa *ALSuite 3*. Para establecer el voltaje se debe ejecutar *Prime95* durante 10 minutos para verificar que el sistema es estable. El voltaje usado por los seis núcleos es de 1,0V. Al escoger la temperatura se debe comprobar que el ventilador es capaz de disipar suficiente calor para tener una temperatura del procesador

El ventilador gira a su máxima velocidad (2000RPM) cuando la temperatura pasa de los 60°C y disminuye su velocidad a 300RPM cuando esta es inferior a los 59°C.

de 60°C. Además, se debe comprobar que al establecer la mínima frecuencia y reducir la velocidad del ventilador al máximo, se genera la temperatura necesaria para alcanzar los 60°C.

El experimento se ejecuta sobre los seis núcleos utilizando 12 hilos independientes de *Prime95* buscando cada uno números primos durante 3 minutos. Las medidas se toman cada segundo, obteniendo 180 muestras en cada punto del experimento. Como muestra más representativa se toma la mediana. Esto evita tomar un espurio como valor de referencia de un punto del experimento. Todos los experimentos de este TFG se han realizado con las tecnologías *Intel Speed Step*, *Speed Shift* y *Turbo* desactivadas. Esto se hace para que el BIOS se haga cargo del control de potencia y rendimiento, evitando así que el SO o el PCU se encarguen de este control (Sec. 5.5).

6.3.2 Resultados

En la figura 6.5 se observa que la potencia medida con el PPA 520 tiene una varianza despreciable. Los cinco espurios que hay pertenecen a las primeras medidas tomadas donde el procesador todavía está en un periodo transitorio ya que no se ha conseguido estabilizar la temperatura. Este diagrama respalda la decisión anterior de tomar la mediana como muestra representativa de cada punto del experimento.

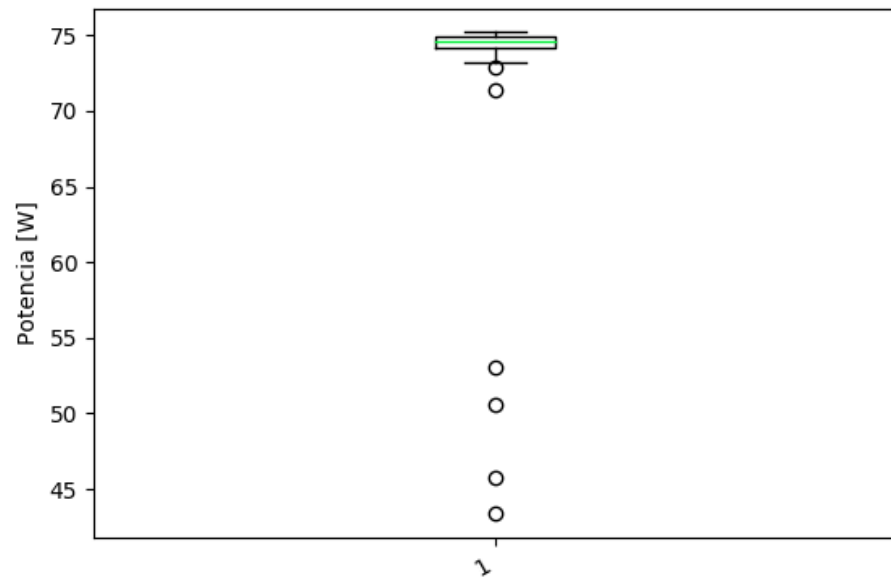


Figura 6.5: Distribución de la potencia medida con el PPA 520 a una frecuencia de 1.4GHz, voltaje del núcleo de 1.0V y temperatura constante a 60°C.

En la figura 6.6 se muestran los puntos obtenidos en este experimento. A partir de estos puntos se hace una regresión lineal y se obtiene la recta $P_{din} = 34,026 * f + 29,4592$. Las medidas se ajustan bastante bien a la recta.

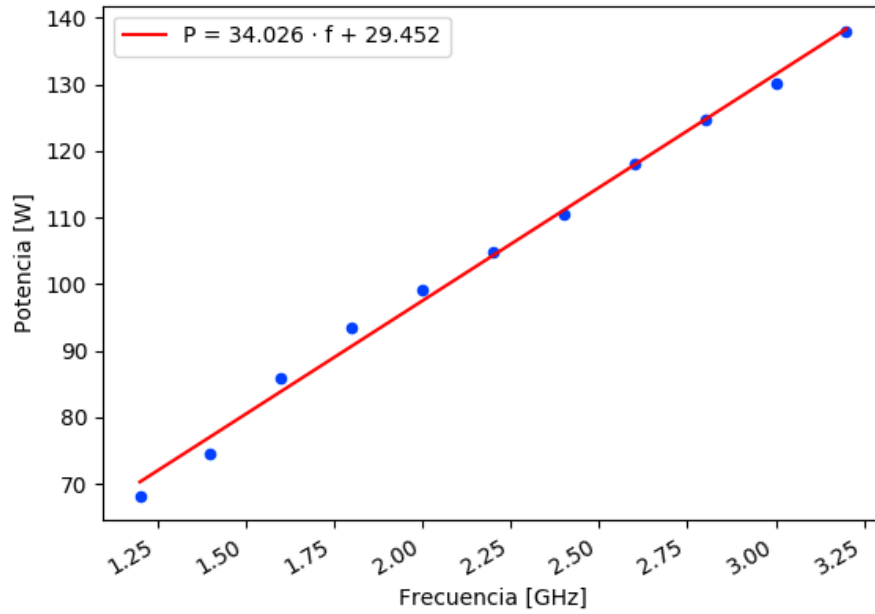


Figura 6.6: Potencia medida con el PPA520 variando la frecuencia entre 1.2GHz y 3.2GHz con incrementos de 200MHz. La temperatura y el voltaje se mantienen constantes a 60°C y 1.0V

En el blog de Henry se realiza este mismo experimento con un procesador de la generación *Sandy Bridge* de 32 nm y otro de 22nm de la generación *Ivy Bridge* [21]. Las relaciones obtenidas entre la potencia dinámica y la frecuencia son $P_{din-Ivy} = 20,1 * f + 0,5659$ y $P_{din-Sandy} = 19,0 * f + 0,0819$. Es decir, la potencia dinámica tiene un factor de incremento mayor en el caso de los procesadores con transistores 2D (*Sandy Bridge*) que en el de los procesadores con transistores 3D (*Ivy Bridge*). Esto es lógico ya que los transistores 3D tienen una menor latencia de conmutación. Comparando nuestro procesador (generación *Skylake*) con los utilizados por Henry se obtiene que la potencia en *Skylake* se incrementa mucho más rápido al aumentar la frecuencia que en los procesadores analizados por Henry (Fig. 6.7). Esto se debe a la composición de cada procesador ya que la capacidad de cada uno es diferente. También influye la actividad del procesador debido a que los experimentos se realizaron con diferentes versiones de *Prime95* que utilizan instrucciones vectoriales distintas.

6.4 EXPERIMENTO 2: POTENCIA Y VOLTAJE

6.4.1 Descripción

El objetivo de este experimento es obtener la relación que existe entre el voltaje y las potencias dinámica (P_{din}) y estática (P_{est}) en la plataforma Asus. Para ello se realizan tres medidas (P_1 y P_2) a temperatura constante (a fin de que no varíe la potencia estática),

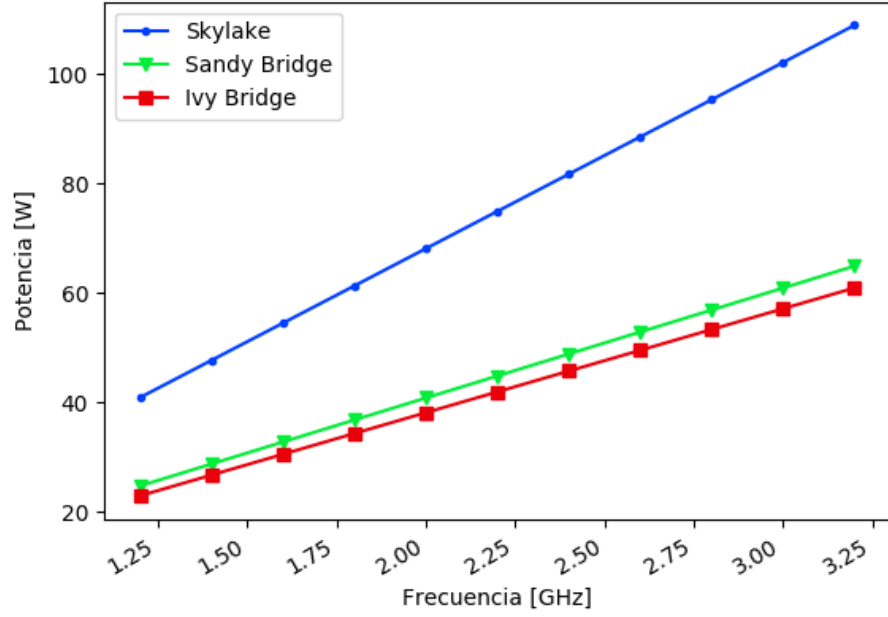


Figura 6.7: Variación de la potencia dinámica al incrementar la frecuencia a temperatura y voltaje fijo. Se compara el procesador Skylake utilizado en este experimento con otros dos de las generaciones *Ivy Bridge* y *Sandy Bridge*.

en las que en consecuencia solo se varía la frecuencia. Considerando la Ec. 2.5 esto da lugar a un sistema de dos ecuaciones (para P_1 y P_2 respectivamente) y dos incógnitas (P_{est} y V), de las que se pueden despejar P_{est} y AC (Ecs. 6.1, 6.2 y 6.3). Los voltajes desaparecen porque las muestras escogidas han sido obtenidas con el mismo voltaje.

$$\Delta P_{din} = P_2 - P_1 = P_{est_2} + ACV_2^2 f_2 - P_{est_1} - ACV_1^2 f_1 \quad (6.1)$$

$$AC = \frac{\Delta P_{din}}{V_2^2 f_2 - V_1^2 f_1} = \frac{\Delta P_{din}}{f_2 - f_1} \quad (6.2)$$

$$\begin{aligned} P_{est} &= P_2 - P_{din} \\ &= P_2 - ACV_2^2 f_2 = P_2 - \frac{\Delta P_{din}}{f_2 - f_1} V_2^2 f_2 \\ &= \frac{P_1 f_2 - P_2 f_1}{f_2 - f_1} \end{aligned} \quad (6.3)$$

Aunque sólo son imprescindibles dos experimentos a dos frecuencias distintas, se decidió realizar tres experimentos para dar más solidez a los resultados. Cada experimento consta de tres fases, cada fase a una frecuencia (2,0GHz, 2,3GHz y 2,6GHz respectivamente), y tomando medidas en 11 puntos diferentes, durante tres minutos en cada uno. La única diferencia entre estos puntos es el voltaje de los núcleos del procesador. El rango de voltajes se varía desde 0,75V hasta 1,1V con incrementos de 35mV. Durante todo el experimento se mantiene una temperatura constante de 65°C.

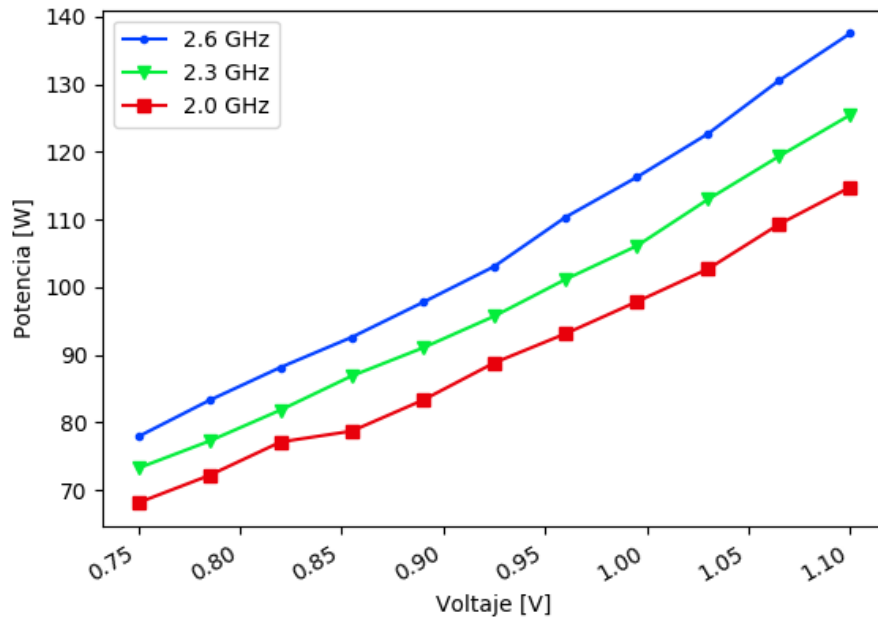


Figura 6.8: Potencia medida con el *PPA520* variando el voltaje entre 0.75V y 1.1V a temperatura constante (65°C) repitiendo el experimento con tres frecuencias: 2.0GHz, 2.3GHz y 2.6GHz

Para elegir el voltaje mínimo (0,75V), se realizó un test de estabilidad del sistema. Para ello, se ejecutaron 12 hilos de *Prime95* durante 10 minutos a la frecuencia máxima del experimento (2,6GHz). Si al final de los 10 minutos no se ha encontrado ningún fallo, se volvía a decrementar el voltaje hasta encontrar el voltaje mínimo para el cuál el sistema es estable. Una vez encontrado el voltaje mínimo, se comprobaba la temperatura máxima que es capaz de generar el procesador con el ventilador a la velocidad mínima con frecuencia y voltaje mínimos (2,0GHz y 0,75V). A continuación, se buscaba el voltaje máximo al cual el ventilador es capaz de mantener el procesador a la temperatura obtenida en el paso anterior. Este último paso se realizaba con una frecuencia de 2,6GHz, que es el punto que más calor genera de todo el experimento. En todos los casos se ha tomado la mediana como muestra representativa de un punto del experimento.

6.4.2 Resultados

En la figura 6.8 se muestra la variación de la potencia total consumida por el procesador al incrementar el voltaje. Se observa que la potencia crece linealmente con el voltaje, lo que se contradice con el modelo de potencia esperado (2.5).

Estos resultados sólo pueden deberse a que no se han conseguido establecer correctamente algunas condiciones del experimento. Para tratar de solucionarlo se repitió este experimento varias veces realizando diferentes cambios.

Primero, se consideró el hecho de que el primer mecanismo de defensa térmica de este tipo de procesadores consiste en la inyección de instrucciones *nop* (de bajo consumo) en el *pipeline*. Esto se podría detectar analizando el número de instrucciones por segundo (IPS). Por ello se recurrió a un análisis de rendimiento utilizando el programa *Intel VTune Amplifier*. Sin embargo, el IPS no cambia en ninguna de las partes del experimento a frecuencia constante, ya que se ejecuta el mismo número de instrucciones independientemente del voltaje del procesador, por lo que se desestimó esta posible causa.

Después, se repitió el experimento ejecutando *Prime95* en un solo núcleo a 50°C, para descartar la posibilidad de que el procesador aplique ahogamiento térmico o de potencia (por rebasamiento de temperatura o de potencia; ver Sec. 5.5.4). Los resultados fueron parecidos a los mostrados en la Fig. 6.8.

En otro experimento, se cambió a o el valor del parámetro *AVX Instruction Core Ratio Negative Offset* desde el BIOS. El valor de esta opción en el experimento mostrado en la figura 6.8 era *Auto*. Esto implica que al ejecutar un programa con muchas instrucciones vectoriales el PCU puede reducir automáticamente la frecuencia y el voltaje de los núcleos del procesador que considere necesario para evitar que este caliente. Esto tendría sentido al ejecutar *Prime95* ya que la mayoría de instrucciones que se ejecutan son vectoriales. El experimento sin embargo volvió a producir resultados aproximables a una función lineal.

También, se midió la potencia consumida por todo el equipo directamente en el enchufe. El objetivo de este experimento es corroborar que el procesador solo se alimenta por las mangueras *EPS12V* y no por la manguera principal que alimenta al resto del equipo (Fig. 6.1). El resultado fue el esperado: la diferencia de potencia entre las medidas del vatímetro conectado a la red eléctrica y el *PPA 520*, que mide la potencia del procesador, es constante durante todo el experimento. Esta diferencia corresponde al consumo de la fuente de alimentación y del resto de componentes de la placa base.

Tras el análisis de resultados de todos los experimentos anteriores, se concluye con la siguiente hipótesis: *el procesador realiza un ajuste autónomo del voltaje aplicado a los núcleos del procesador en ciertas condiciones independientemente de la configuración especificada en el BIOS o programa AISuite 3*. De hecho, en el manual de Intel para desarrolladores software [28] se especifica que el PSU podría ignorar una orden del SO para no entrar en un estado de reposo (estado C) debido a que no es eficiente, como ya describimos en la Sec. 5.5. Algo parecido podría estar ocurriendo en este caso cuando el BIOS le pide al PCU que aplique un voltaje especificado por el usuario. Este podría ignorar la orden del BIOS al considerar que ese nuevo voltaje es ineficiente energéticamente. La diferencia de este experimento con el caso que se recoge en el manual de Intel es que no se utiliza el estándar ACPI (no hay estados

C) y que las órdenes del control de potencia y rendimiento lo hace el BIOS a través del bus PECCI y no el SO.

6.5 EXPERIMENTO 3: POTENCIA ESTÁTICA Y TEMPERATURA

6.5.1 Descripción

Este experimento persigue determinar la variación de la potencia estática con el cambio de temperatura del procesador. Se ha fijado una frecuencia de 2.0GHz y un voltaje de 1.0V. En cada punto del experimento, primero se regula la temperatura con la aplicación *AISuite 3*. Después se lanza *Prime 95* con 12 hilos independientes mediante el interfaz gráfico. Tras esto, se espera un tiempo para que la temperatura del procesador se estabilice y se inician los programas de recolección de datos. En cada punto del experimento se toman 180 medidas durante tres minutos. Al terminar un punto del experimento, se apagan solo los programas de recolección de datos y se vuelve a regular la velocidad del ventilador para el siguiente punto. La diferencia de temperatura entre cada punto es de 5°C. El experimento finaliza a los 75°C ya que a partir de esta temperatura es probable que el procesador realice ahogamiento térmico y esto influiría en las medidas de potencia.

6.5.2 Resultados

La potencia estática es proporcional al inverso de una función exponencial con la temperatura en el exponente:

$$P_{est} \propto e^{(-qV_{th}/kT)} \quad (6.4)$$

Es decir, la forma de la función es parecida al de un logaritmo. Dado que la potencia dinámica no varía, el cambio en la potencia observado en el experimento se debe únicamente a la potencia estática.

Los puntos medidos en el experimento se aproximan sin embargo a una función lineal. Esto sucede porque las temperaturas medidas no son tan extremadamente altas como para ver llegar a ver la zona asintótica. Para alcanzar esta zona se tendría que probar a deshabilitar todas las medidas de protección relacionadas con la temperatura (Sec. 5.5.1). Esto permitiría realizar el experimento a mayores temperatura sin ahogamiento térmico, aunque asumiendo un riesgo muy probable de estropear el procesador.

6.5.3 Conclusiones

El estudio de la gestión de potencia y rendimiento de esta plataforma ha supuesto un gran reto debido a la poca documentación

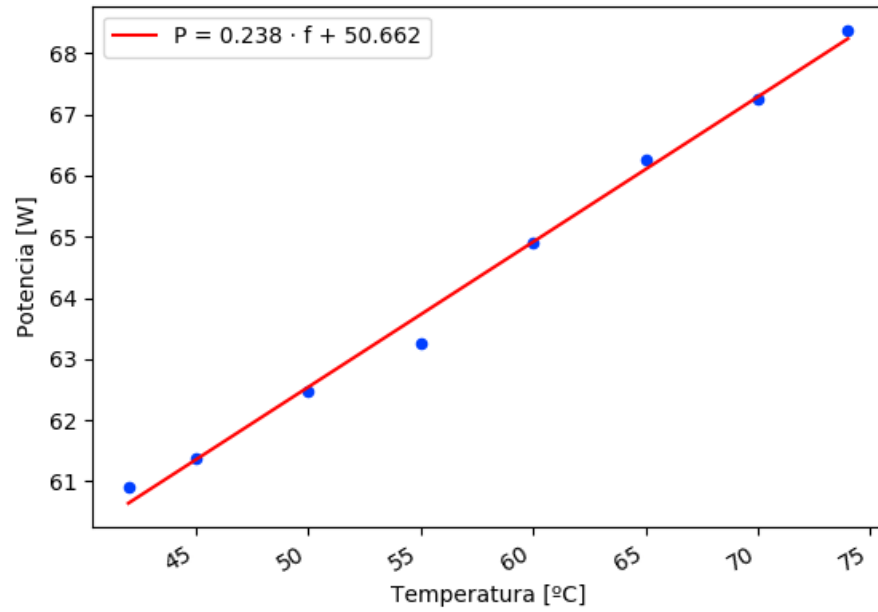


Figura 6.9: Variación de la potencia estática medida con el *PPA520* al cambiar la temperatura. Se varía la temperatura entre 42°C y 75°C. El voltaje y la frecuencia están fijos a 1.0V y 2.0GHz.

disponible. Los resultados obtenidos en el segundo experimento (Sec. 6.4) evidencian la existencia de un mecanismo autónomo de control térmico o de consumo que se activa cuando los ajustes establecidos por el usuario o el *SO* son inefficientes.

CONSIDERACIONES FINALES

7.1 CONCLUSIONES

En la primera parte del proyecto se ha instrumentado un entorno experimental para obtener el comportamiento térmico y energético de una Raspberry Pi. Además, se detallaron los pasos y herramientas necesarias para replicar el estudio realizado. También se documentaron los planos de control de voltajes y frecuencias de la placa junto con la gestión de potencia y rendimiento. Al analizar los resultados descubrimos las limitaciones que este equipo tiene y sus implicaciones en los resultados.

En la segunda parte del proyecto se hizo un análisis del mercado de las placas base identificando las características recomendables para realizar *overclocking*. Después, se configuró y montó el equipo adquirido, y se configuró y documentó el entorno experimental necesario para modelar su comportamiento energético y térmico. A continuación, se documentó la gestión de potencia y rendimiento de la plataforma combinando información de diferentes fuentes dado que la documentación proporcionada por los fabricantes es muy escasa. Este estudio permite conocer las implicaciones de las diferentes tecnologías utilizadas en el comportamiento de la plataforma.

Con los experimentos realizados en la plataforma Asus se descubrió que la potencia se incrementa linealmente con el voltaje, y no cuadráticamente como se esperaba a partir del modelo habitual de potencia. Esto se debe a que la unidad de control de potencia del procesador hace un ajuste autónomo del voltaje en caso de encontrar una configuración más eficiente que la impuesta por el [SO](#) o el [BIOS](#).

La principal conclusión de este trabajo es que incluso con un equipamiento altamente configurable, pueden existir mecanismos de control térmico o de consumo que alteren las medidas de potencia, cuya posibilidad de gestión habrá que seguir estudiando, incluyendo la posibilidad de recurrir a otros modelos de placa o procesador.

7.2 TRABAJO FUTURO

El trabajo realizado hasta el momento abre varias vías de continuidad que exponemos a continuación.

- Adquirir un nuevo procesador Intel de la familia *Skylake-X Refresh* y repetir los experimentos en la misma placa base. Con esto se podría obtener una comparación del comportamiento energético y térmico entre dos procesadores muy similares con dos

años de diferencia. En este nuevo proyecto bastaría con montar el nuevo procesador y seguir las instrucciones del Cap. 6.

- Evaluación de algoritmos de planificación tiempo real para multiprocesadores sensible a temperatura y energía en la plataforma Asus basándose en la caracterización térmica y energética obtenida en este TFG. Estos resultados servirían para validar los diferentes algoritmos desarrollados por el Grupo de Arquitectura de Computadores de la Univ. de Zaragoza (GaZ) y el grupo de Ingeniería de Control de la Unidad Guadalajara del Centro de Investigaciones y Estudios Avanzados (CINVESTAV) de México.
- Desarrollo de un controlador para Linux que ofrezca una interfaz parecida al de la aplicación *AlSuite 3*.
- Desarrollo de un controlador para el módulo *CPUFreq* de Linux que permita realizar un control personalizado de la potencia y rendimiento.
- Adquisición y montaje un sistema de refrigeración líquida en la plataforma Asus. Esto ofrecería un control de temperatura más preciso y con mayor poder de refrigeración que el realizado en este TFG dado que la capacidad de absorción del agua es mayor que la del aire. Por tanto, se podrían repetir todos los experimentos con unos rangos de frecuencias y voltajes más amplios y obtener así una caracterización aún más completa.
- Realizar ingeniería inversa al programa *AlSuite 3* para obtener más información del control de potencia y rendimiento realizado por la plataforma Asus. Esto permitiría averiguar de forma más precisa la comunicación entre BIOS y PCU (Sec. 5.5)
- Construir un ventilador regulable por PWM e implementar un módulo para el control de temperatura para la Raspberry Pi [2].

ANEXOS

A.1 DIAGRAMA GANTT DEL PROYECTO

En esta sección se muestra el diagrama de Gantt del proyecto. En él no están incluidas las horas de trabajo previo, dedicadas a la parte de la Raspberry, en la asignatura de *Laboratorio de Sistemas Empotrados*

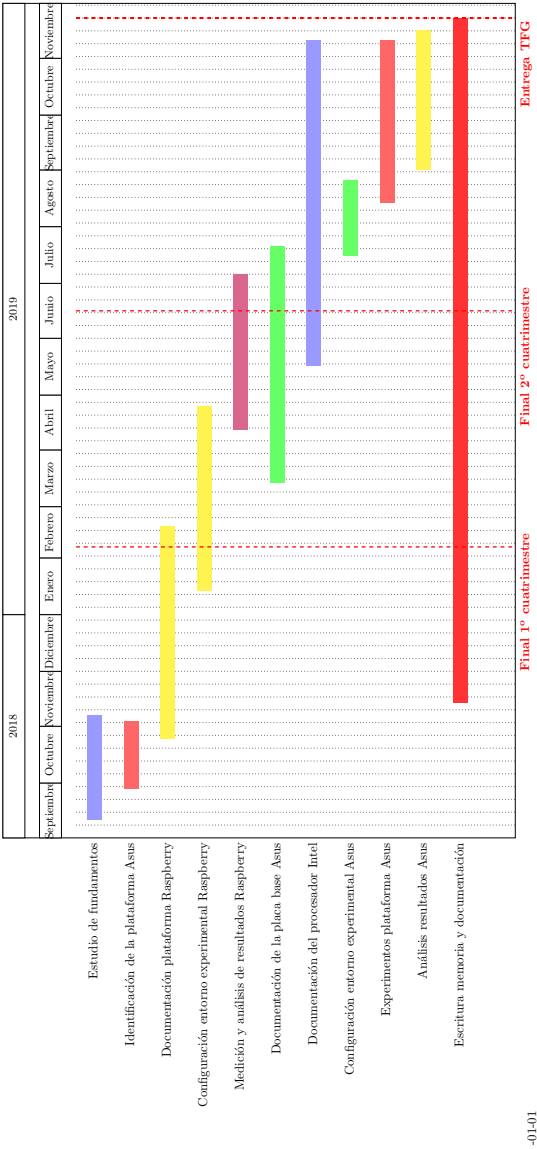


Figura A.1: Diagrama de Gantt

A.2 VELOCIDAD DE CONMUTACIÓN MOSFET

Los transistores **MOSFET** tienen tres zonas de funcionamiento: corte, óhmica y saturación. Cuando está en corte no hay conducción entre el drenador y la fuente, mientras que en las otras dos sí que conduce. Por tanto, un transistor que está conduciendo para dejar de hacerlo, debe salir de la zona de saturación, pasar por la óhmica y llegar a la zona de corte. Este tiempo de conmutación viene marcado por el tiempo de carga o descarga del condensador que está conectado a la salida del transistor (Fig. 2.2).

Las condiciones de conducción entre el drenador y la fuente en las tres zonas de funcionamiento son las siguientes:

- Corte:

$$I_{DS} = 0A$$

- Óhmica:

$$I_{DS} = k(V_{GS} - V_{th})^2$$

- Saturación:

$$R_{DS} = \frac{1}{V_{GS} - V_{th}}$$

donde I_{DS} representa la intensidad entre el drenador y la fuente, V_{GS} la tensión entre la puerta y fuente y k es una constante que depende de la anchura del canal entre drenador y fuente. V_{th} es la tensión umbral que se debe superar para que un transistor NPN empiece a conducir.

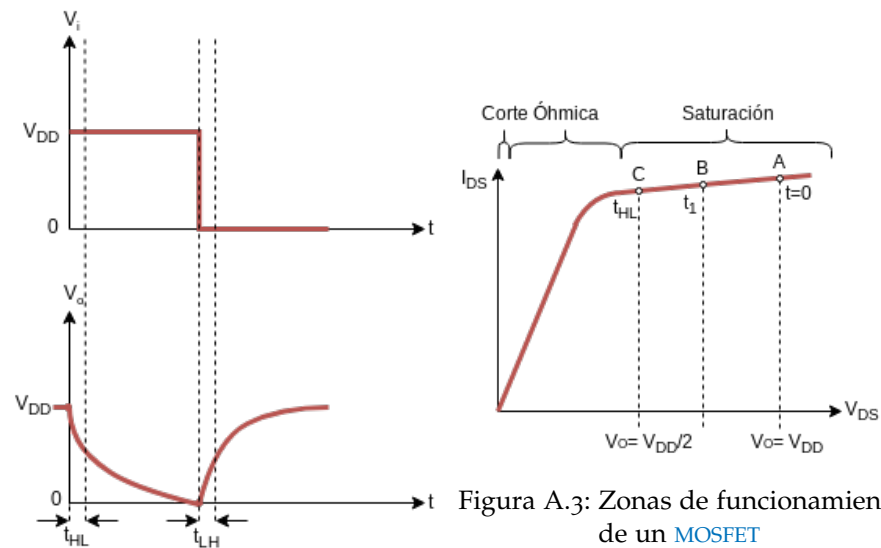


Figura A.3: Zonas de funcionamiento de un **MOSFET**

Figura A.2: Tiempo de conmutación en un **MOSFET**

Partiendo de un transistor que se encuentra en saturación en el punto A de la Fig. A.3 ($t = 0$ y $V_{DS} = V_{DD}$) con una corriente de

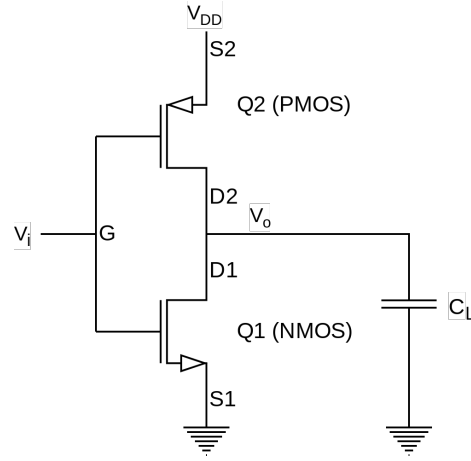


Figura A.4: Esquema puerta NOT con dos MOSFET

conducción I_{DS} del transistor Q_1 , se obtiene que la tensión de salida de la puerta NOT (Fig. A.4) es:

$$V_o = V_{DD} - \frac{Q}{C_L} = V_{DD} - \frac{tI_{DS}}{C_L} = V_{DD} - \frac{kt(V_{DD} - V_{th})^2}{C_L} \quad (A.1)$$

siendo Q la carga perdida y C_L la capacidad del condensador en un instante t .

Cuando $V_o = V_{DS1} = V_{GS1} - V_{th}$ (Fig. A.3, punto B), se pasa a la zona óhmica. El tiempo necesario para que esta condición se cumpla viene determinado por t_1 :

$$t_1 = \frac{V_{th}}{k(V_{DD} - V_{th})^2} * C_L \quad (A.2)$$

A partir del instante $t = t_1$, la descarga del condensador se realiza como un circuito compuesto por una resistencia (r_{DS1}) y un condensador. Para hallar el tiempo restante t_2 en el que la salida de la puerta NOT pasa de un valor lógico 1 a 0, hay que despejar t_2 de la ecuación A.3

$$V_o = V_0(t = t_1)e^{\frac{-(t-t_1)}{\tau}} = (V_{DD} - V_{th})e^{\frac{-t_2}{\tau}} = \frac{V_{DD}}{2} \quad (A.3)$$

con

$$\tau = r_{DS1}C_L = \frac{C_L}{k(V_{DD} - V_{th})} \quad (A.4)$$

$$t_2 = -\frac{C_L}{k(V_{DD} - V_{th})} \ln \frac{V_{DD}}{2(V_{DD} - VT)} \quad (A.5)$$

El tiempo de propagación de la puerta NOT de nivel alto a bajo o viceversa viene determinado por:

$$t_{HL} = t_{LH} = t_1 + t_2 = \frac{C_L}{k(V_{DD} - V_{th})} \left(\frac{V_{th}}{V_{DD} - V_{th}} - \ln \frac{V_{DD}}{2(V_{DD} - V_{th})} \right) \quad (A.6)$$

De la ecuación A.6 se observa que para conseguir una alta velocidad de conmutación se necesita una pequeña tensión de umbral y capacidad del condensador adyacente, mientras que la tensión de alimentación y la constante de transconductancia k deben ser lo más altas posible. Relacionando la ecuación A.6 con la frecuencia de un circuito integrado se obtiene que:

$$F \propto \frac{(V_{DD} - V_{th})^2}{V_{DD}} \quad (A.7)$$

Respecto a la potencia dinámica, es proporcional a la actividad (número de puertas que conmutan simultáneamente, A) y la carga capacitiva del circuito integrado. También hay que tener en cuenta una pequeña pérdida debida a la fuga por cortocircuito entre fuente y masa mientras el transistor está en zona la óhmica, aunque está se puede considerar despreciable en la mayoría de los casos.

$$P_{din} = ACV^2F + t_{HL}AVI_{short} \simeq \frac{1}{2}CV^2F \quad (A.8)$$

A.3 GESTIÓN DE ENERGÍA Y RENDIMIENTO EN LINUX

El escalado de rendimiento de CPU en Linux se realiza por medio de un subsistema llamado **CPUFreq** (*CPU Frequency Scaling*) que está compuesto por tres capas: núcleo (*core*), gobernadores (*scaling governors*) y drivers de escalado (*scaling drivers*). El **CPUFreq core** proporciona el bloque necesario para comunicarse con el scheduler y gestionar los diferentes gobernadores. También es capaz de comunicarse con la interfaz de usuario desde la cuál se puede modificar y consultar el estado del control de rendimiento (*performance scaling*). La principal función de esta parte es independizar el control de potencia y rendimiento del sistema de la plataforma hardware.

Los *scaling governors* son los algoritmos o políticas específicos de cada sistema que establecen la relación entre el rendimiento y consumo energético de un sistema. Alguno de ellos usan parámetros como el nivel de batería o la carga de trabajo del equipo. Los gobernadores disponibles en el **SO Linux** utilizado en la Raspberri Pi en este **TFG** son: *conservative*, *ondemand*, *userspace*, *powersave*, y *performance*.

Por último, el scaling driver es el firmware encargado de realizar los cambios en el hardware para que la elección tomada por la política

de gestión de energía tenga efecto. Los drivers son dependientes del sistema y solo se permite tener registrado/activo uno de ellos[58].

A continuación, se explica el funcionamiento de [CPUFreq](#) en Raspbian, el [SO](#) instalado en la Raspberry utilizada en este [TFG](#).

Durante el proceso de arranque el cargador del sistema o *bootloader* establece una tensión y frecuencia por defecto. Tras cargar los módulos más importantes del sistema, como el driver de la sd, se carga un módulo con el controlador de energía usado por el equipo. A continuación, se crea el `struct cpufreq_driver` y se completa con características del sistema relacionadas con control de potencia y rendimiento. Por ejemplo, contiene una tabla con las frecuencias disponibles, la latencia de transición de frecuencia o la frecuencia actual. En el caso de la Raspberry Pi 3B, las frecuencias máximas y mínimas con las que se completa este struct están especificadas en el fichero `/boot/config.txt` bajo las variables `arm_freq` y `arm_freq_min`, respectivamente.

Al invocar la función `cpufreq_generic_init(struct cpufreq_policy *policy...)` se completa el `struct cpufreq_policy` con la información relativa al hardware proporcionada por el driver. Cada conjunto de [CPUs](#) que compartan algún registro hardware relacionado con el control de potencia y rendimiento estará representado por un único `struct cpufreq_policy`. Este *struct* contiene la máscara de [CPUs](#) a las que representa, las frecuencias soportadas, etc. En el siguiente paso, CPUFreq crea el interfaz que permite al usuario modificar y consultar los estados de consumo y rendimiento del sistema. Para ello, en Raspbian se crea un `kobject` en el directorio `/sys/devices/system/cpu/`. Posteriormente se añade a la estructura `cpufreq_policy` un puntero al gobernador utilizado por defecto. De esta manera al cambiar de gobernador desde espacio de usuario, solo hay que invocar una función que cambie el gobernador apuntado por dicho campo. Una vez iniciado el gobernador, el planificador o *scheduler* se encarga de la actualización periódica de los gobernadores invocando a la función `handle_update()` del subsistema [CPUFreq](#). A su vez, este invoca al gobernador correspondiente, que invoca al driver para aplicar los cambios necesarios en el hardware.

Los gobernadores disponibles en Raspbian son:

- **performance:** el procesador siempre opera a la máxima frecuencia
- **powersave:** el procesador siempre opera a la mínima frecuencia
- **userspace:** el usuario elige una de las frecuencias posibles del procesador escribiendola en el fichero `/sys/devices/system/cpu/cpufreq/cpuX/policy0/scaling_setspeed`

- **ondemand:** la frecuencia se establece según el ratio de utilización del procesador en el periodo anterior. El tiempo transcurrido entre actualizaciones se puede modificar desde el espacio de usuario. También se puede establecer un umbral de utilización a partir del cuál la CPU en cuestión pase a ejecutarse a su máxima frecuencia.
- **conservative:** es muy parecido al gobernador *ondemand* pero, aquí los cambios son menos agresivos dado que se realizan pequeños incrementos o decrementos de la frecuencia. Además, el periodo en actualizaciones del estado energético del procesador es más grande.

En la figura A.5 se muestra un diagrama con los módulos principales de **CPUFreq**. También hay un resumen de la pila de invocaciones realizadas durante el inicio del **SO** (flechas 1-4) y la actualización periódica del estado de energía y rendimiento (flechas 5-7.3).

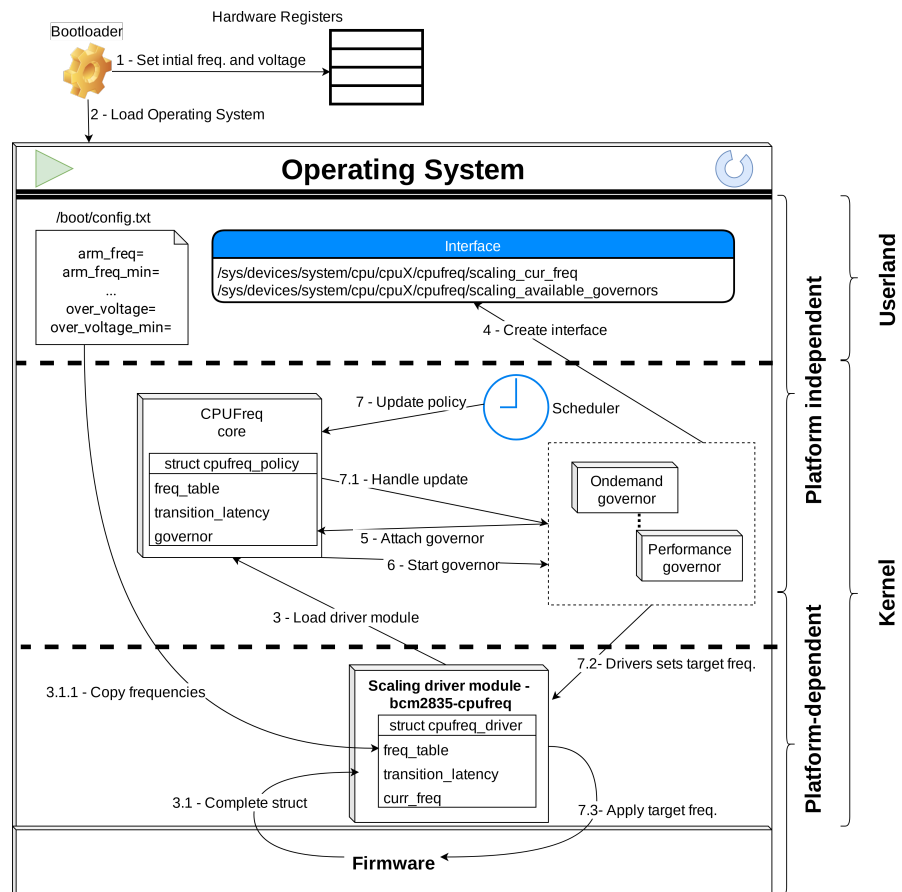


Figura A.5: Diagrama de bloques y ejecución de **CPUFreq** en la Raspberry Pi 3B

A.4 ADVANCED CONFIGURATION AND POWER INTERFACE

El *Advanced Configuration and Power Interface* o **ACPI** es un estándar definido por empresas como Intel o Hewlett-Packard para establecer una interfaz común para el control de potencia y rendimiento de una máquina. Esta interfaz se define tanto desde el punto de vista del hardware como del software.

Los principales componentes del estándar **ACPI** son los registros, el firmware y las tablas de descripción del sistema (Fig. A.6). En las tablas **ACPI** de descripción del sistema se describe en lenguaje máquina de **ACPI** (**AML**) el interfaz hardware. Por ejemplo, estas tablas contienen los pares de frecuencia y voltaje permitidos por el procesador. Los registros **ACPI** son los registros del equipo que se deben modificar en el control de potencia y rendimiento. Por último, el firmware **ACPI** es un código que, basándose en los registros y tablas **ACPI**, implementa acciones básicas de control como el arranque, apagado o cambio de estado de potencia y rendimiento de un componente dado.

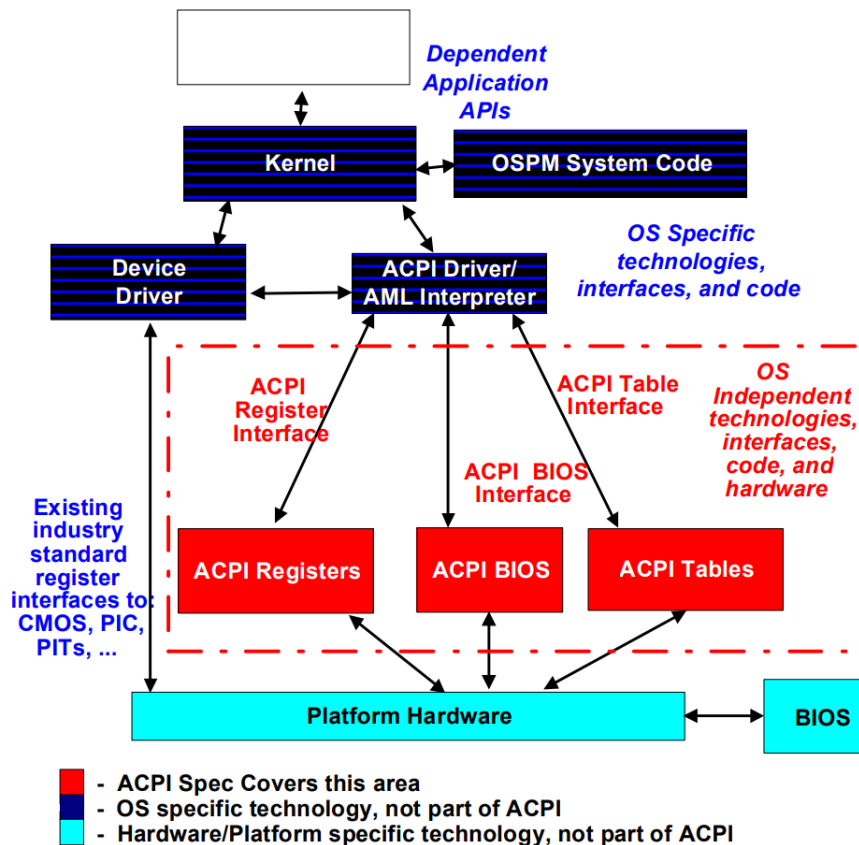


Figura A.6: Relación entre los principales componentes del estándar ACPI.
Fuente: [9].

Para llevar a cabo este control, el subsistema del **SO** encargado del control de potencia y rendimiento (**OSPM**) se comunica con un controlador que interpreta el código **AML** de las tablas **ACPI** y proporciona la

información necesaria al [SO](#) para que realice dicho control. Después, el firmware [ACPI](#) aplica los cambios especificados por el [OSPM](#). La implementación del intérprete y el firmware que aplica los cambios en el hardware se suele hacer en un mismo controlador o *driver*.

La interfaz definida por [ACPI](#) permite controlar la energía consumida por cada componente que conforma un equipo informático. Para ello, primero se definen los siguientes estados de funcionamiento global del sistema:

- *G3 - Mechanical Off*: el equipo no está conectado a una fuente de energía eléctrica.
- *G2 - Soft Off*: estado de larga duración en el que el hardware no ejecuta código.
- *G1 - Sleeping*: estado en el que no se ejecuta código del usuario. Por ejemplo, al suspender un equipo se entra en este estado.
- *G0 - Working*: estado de funcionamiento normal del sistema en el que se ejecuta código del usuario y [SO](#).
- *Legacy*: se entra en este estado cuando algún componente no es compatible con el estándar [ACPI](#).

Ciertos componentes como los discos duros se pueden controlar de forma individual mediante los estados D:

- *D3 - Off*: apagado
- *D3 - hot*: estado de ahorro energético en el que se almacena parte del contexto del componente. En ciertas ocasiones, puede que sea necesario un reinicio del dispositivo para su correcto funcionamiento.
- *D2*: estado de menor ahorro energético que el *D3* pero en el que se guarda más información del contexto.
- *D1*: estado de reposo en el que menos energía se ahorra y más información del contexto se almacena. Para volver al estado *Do* nunca es necesario un reinicio del componente.
- *Do - Fully-On*: componente en pleno funcionamiento.

Además, el estado *G1* se divide en otros cinco estados donde *S1* corresponde al estado menos profundo ya que el equipo consume mucha más energía que en el estado *S5*.

En el estado *G0* se definen los estados C de reposo para el procesador (Fig. [A.7](#)):

- *C0*: el procesador ejecuta instrucciones
- *C1*: el procesador ya no ejecuta instrucciones

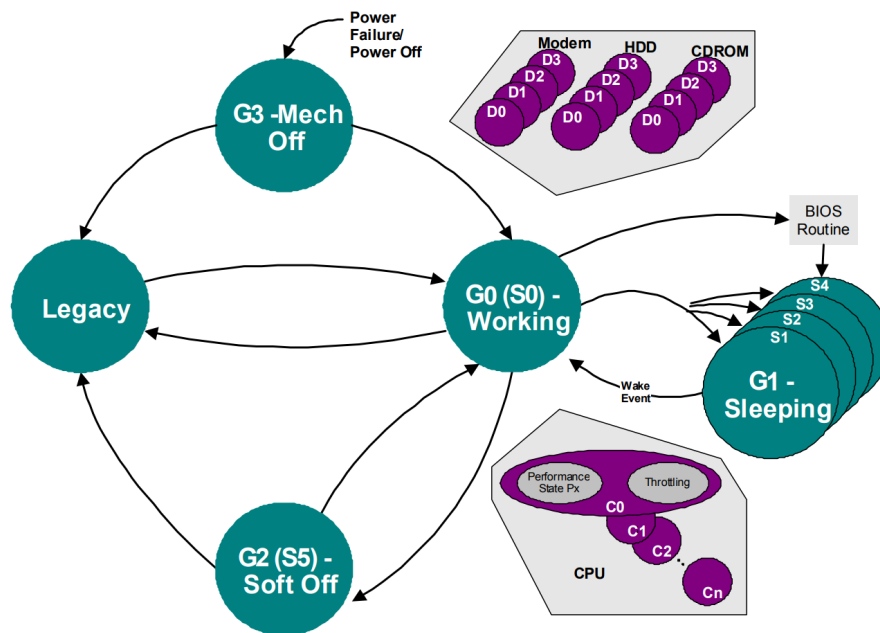


Figura A.7: Estados ACPI y sus relaciones. Fuente: [9].

- C2: estado de mayor ahorro energético y con una mayor latencia de transición al estado C0.
- C3: en este estado las caches L3 mantienen el estado pero ya no realizan *snooping*.

Estos estados se definen a nivel de procesador (estados *Package C* o *PC*) y de núcleo (estados *Core C* o *CC*). El estado del procesador lo determina el estado del núcleo de menor profundidad. Es decir, si un núcleo se encuentra en el estado CC1 y el resto en el CC3 el estado del procesador será PC1.

A su vez, el estado C0 se divide en estados P o *performance*. El número de estados P depende del modelo del procesador y también están definidos a nivel de núcleo. P1 corresponde al estado de máximo rendimiento y consumo energético mientras que en el estado Pn se obtiene un rendimiento bajo.

Por último, en el estado C0 existen unos estados especiales (estados T) de baja potencia a los que accede cuando el procesador supera unos umbrales de temperatura y se aplica el ahogamiento térmico.

A.5 ELECCIÓN PROGRAMA PARA ESTRESAR EL PROCESADOR

Los candidatos a ser elegidos como programa de estrés del procesador para este proyecto son: *Prime95*, *AIDA64*, *Intel Extreme Tuning Utility* y *OCCT*. La prueba realizada consiste en ejecutar cada programa y analizar los consumos de potencia.

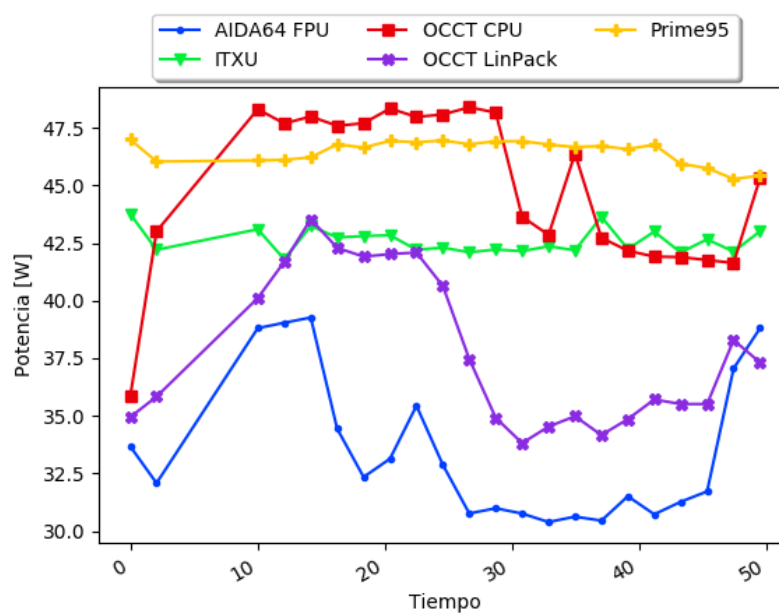


Figura A.8: Comparación de potencia por los siguientes programas de estrés: *Prime95*, *AIDA64*, *Intel Extreme Tuning Utility* y *OCCT*

A partir de la gráfica A.8 se obtiene que el mejor programa para este caso es Prime95 dado que es el que más potencia media consume y además los hace de manera constante.

A.6 BIOS

A continuación, se muestra un lista con los apartados y opciones (en azul) más relevantes que se pueden modificar desde la BIOS de la placa ASUS RAMPAGE VI EXTREME OMEGA:

- CPU Configuration
 - CPU Power Management Configuration
 - [Autonomous Core C-State](#): estados de reposo o bajo rendimiento
 - [Intel Speed Shift Technology](#)
 - [MFC Mode Overirde](#): si se activa se podrá descativar el Turbo 3.0 desde la aplicación de Intel. En caso contrario, los estados turbo son gestionados por el [SO](#).
 - [Active Processor Cores](#): permite apagar un núcleo por completo
 - [Maximum CPU Core Temperature](#): temperatura máxima permitida por un core.

- Extreme Tweaker: desde esta sección se realizan la mayoría de controles usados para el *overclocking*.
 - [Ai Overclock Tuner](#): manual o auto. Se utiliza para ajustar el BCLK. En modo manual se especifica la frecuencia del bus con una precisión de 1MHz (por defecto son 100MHz).
 - [ASUS MultiCore Enhancement](#): la optimización turbo se realiza según las heurística de ASUS. Si se desactiva esta opción, los estados Turbo son gestionados por Intel.
 - [AVX Instruction Core Ratio Negative Offset](#)
 - [AVX-512 Instruction Core Ratio Negative Offset](#)
 - [CPU Core Ratio](#): frecuencia de los núcleos. En auto la frecuencia se ajusta según las condiciones del equipo, con *Sync All Cores* el usuario establece la misma frecuencia a todos los núcleos y *by core usage* permite ejecutar cada núcleo a la frecuencia y tensión especificado por el usuario.
 - [Min, Máx CPU Cache Ratio](#): frecuencias máximas y mínimas de la caché L3.
 - [CPU SVID Support](#): si se desactiva esta opción el procesador ya no comunica a los [VRM](#) el voltaje necesario.
 - DRAM Timing Control: panel de control y monitorización de los ciclos de latencia de la DRAM
 - External Digi+ Power Control: desde este menú se pueden ajustar parámetros relacionados con los reguladores de la fuente como el LLC, la corriente máxima a través del canal que alimenta el procesador, etc. También se puede ajustar la frecuencia de los [VRM](#), los voltajes de arranque de los principales dominios eléctricos procesador (núcleos, sistema agente y entrada/salida).
 - CPU Power Management: este panel sirve para ajustar parámetros relacionados con el control de potencia interno del procesador.
 - [Enhanced Intel SpeedStep Technology](#)
 - [Turbo Mode](#)
 - [CPU Integrated VR Current Limit](#): intensidad máxima de los reguladores de tensión integrados. Aumentando este valor se evitan iniciar el proceso de ahogamiento de potencia debido a que la potencia consumida durante un periodo de tiempo es demasiado alta.
 - [CPU Integrated VR Fault Management](#)
 - [CPU Integrated VR Efficiency Management](#): regula la eficiencia de los reguladores integrados.

- Tweaker Paradise: apartado con las configuraciones de los [PLLs](#) y voltajes del controlador de memoria.
- [CPU Input Voltage](#): voltaje del raíl principal (V_{CCIN}) que alimenta el procesador. Desde la [BIOS](#) permite un rango entre 0.8V y 2.7V, aunque los valores de funcionamiento en condiciones normales están entre 1.55V y 1.80V.
- [DRAM Voltage](#): voltaje de los railes que alimentan dos controladores de memoria.
- [CPU VCCIO Voltage](#): voltaje del raíl que suministra energía al dominio eléctrico de la entrada/salida del procesador.
- [CPU System Agent Voltage](#): tensión de entrada al sistema agente.
- [Uncore Voltage Offset](#): establece un offset respecto a la tensión nominal de otros dominios internos del procesador. Por ejemplo, la Unidad de Control de Potencia ([PCU](#)).
- [CPU Core voltage](#): voltaje que alimenta los cores. Este atributo no permite regular la tensión de cada núcleo por separado.
- [CPU cache Voltage](#): voltaje suministrado a la caché L3.
- [PCH Core Voltage](#): tensión de funcionamiento del [PCH](#).
- [PLL voltage](#): voltaje del [PLL](#) principal que genera el reloj del sistema.

A.7 PRINCIPALES CARACTERÍSTICAS DEL PROCESADOR INTEL UTILIZADO

Las características más relevantes del procesador Intel Core i7-7800X utilizado en este proyecto son las siguientes:

- N° núcleos: 6, con 2 hilos por núcleo
- Frecuencia base: 3,5GHz
- Frecuencia máxima: 4,0GHz
- Chipset: *FCLGA2066*
- Caché L3 compartida: 8,25MB
- TDP: 140W
- Temperatura máxima: 100 °C
- Canales de memoria: 4
- Tamaño máximo de memoria: 128GB
- Tipo de memoria: *DDR4 – 2400*

- Tecnologías para el control de energía y rendimiento: Intel Turbo Boost Max 3.0, Intel Speed Step e Intel Speed Shift
- Extensiones vectoriales: *Intel SSE4.1, Intel SSE4.2, Intel AVX2, Intel AVX-512 e Intel AVX-512 FMA*

A.8 FUENTE DE ALIMENTACIÓN

La elección de la fuente de alimentación o PSU se realizó en función de la potencia máxima requerida, estimada en unos 300W. Se optó por una fuente de alimentación *RMx Series RM750x*, que suministra una potencia máxima de 750W, para posibilitar futuras ampliaciones. Es una fuente modular con una eficiencia del 93 % para 400W y 92 % para 250W. Tiene dos puertos para conexiones EPS12V mediante los que se alimenta al procesador, uno de 24 pines para la placa base, cuatro para componentes instalados en el PCIe y nueve para conectores SATA.

A.9 REFRIGERACIÓN

El ventilador instalado es un *Noctua NH-U9S*. Tiene un disipador de aluminio de 807cm^3 y 10 caleoductos [41]. El ventilador tiene un amplio rango de velocidades ($400\text{rpm} - 2000\text{rpm}$) y un caudal máximo de $79\text{m}^3/\text{h}$. Por el momento, solo hay acoplado un ventilador pero existe la posibilidad de añadir uno más en el otro extremo del disipador.

En la revisión hecha en bit-tech.com se configura un equipo con el mismo procesador y una placa base de Asus muy parecida [35]. Dicho equipo con el procesador a 4.8GHz y 1.27V consumía 99W en reposo y 285W a plena usando el programa Prime95.

BIBLIOGRAFÍA

- [1] James Adams. *More power to your Pi*. 2018. URL: <https://www.raspberrypi.org/blog/pi-power-supply-chip/> (visitado 04-09-2019).
- [2] Aerandir14. *PWM Regulated Fan Based on CPU Temperature for Raspberry Pi*. 2018. URL: <https://www.instructables.com/id/PWM-Regulated-Fan-Based-on-CPU-Temperature-for-Ras/> (visitado 04-09-2019).
- [3] Asus. *Asus Rog Rampage VI Extreme Omgea*. 2017. URL: <https://www.asus.com/es/Motherboards/ROG-RAMPAGE-VI-EXTREME-OMEGA/specifications/> (visitado 17-10-2019).
- [4] Asus. *[Motherboard] AI Suite 3 - Introducción*. 2018. URL: <https://www.asus.com/es/support/FAQ/1012780/> (visitado 04-09-2019).
- [5] Steven Bassiri. *Tweaktown's Ultimate Intel Skylake Overclocking Guide*. 2015. URL: <https://www.tweaktown.com/guides/7481/tweaktowns-ultimate-intel-skylake-overclocking-guide/index5.html> (visitado 04-09-2019).
- [6] Steven Bassiri. *ASUS Rampage VI Extreme Omega Motherboard Review*. 2019. URL: <https://www.tweaktown.com/reviews/8966/asus-rampage-vi-extreme-omega-motherboard-review/index.html> (visitado 04-09-2019).
- [7] Kris Boughton. *Overclocking Intel's New 45nm QX9650: The Rules Have Changed*. Dic. de 2007. URL: <https://www.anandtech.com/show/2404/5> (visitado 04-09-2019).
- [8] Broadcom. *BCM2835 ARM Peripherals*. 2012. URL: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/BCM2835-ARM-Peripherals.pdf> (visitado 04-09-2019).
- [9] Hewlett-Packard Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies Ltd. y Toshiba Corporation. «Advanced Configuration and Power Interface Specification». En: 6 (jun. de 2011), págs. 1-54.
- [10] Intel Corporation. *Procesador Intel® Core™ i7-7800X serie X*. 2017. URL: <https://ark.intel.com/content/www/es/es/ark/products/123589/intel-core-i7-7800x-x-series-processor-8-25m-cache-up-to-4-00-ghz.html> (visitado 17-10-2019).
- [11] Richtek Technology Corporation. *RT8088A*. 2013. URL: https://www.richtek.com/assets/product_file/RT8088A/DS8088A-00.pdf (visitado 04-09-2019).

- [12] *CPU Monitoring With DTS/PECL*. 322683. Ver. 001. Intel® Corporation. Sep. de 2010.
- [13] Ian Cutress. *The Intel Skylake Mobile and Desktop Launch, with Architecture Analysis*. 2015. URL: <https://www.anandtech.com/show/9582/intel-skylake-mobile-desktop-launch-architecture-analysis/7> (visitado 04-09-2019).
- [14] Frank Dehmelt. *Adaptive (Dynamic) Voltage (Frequency) Scaling — Motivation and Implementation*. 2014. URL: <http://www.ti.com/lit/an/slva646/slva646.pdf?DCMP=fullyc&HQS=pwr-alps-lpdc-fullyc-fr59x69x-20140624-dvs-mcs-en> (visitado 04-09-2019).
- [15] EfectoLed. *Repartidor de Energía de 2P - efectoLed*. URL: <https://www.efectoled.com/es/comprar-inicio-generico/2773-repartidor-de-energia-de-2p.html> (visitado 04-09-2019).
- [16] *Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor*. 301170. Ver. 001. Intel® Corporation. Mar. de 2004.
- [17] Raspberry Pi Foundation. *Overclocking options in config.txt*. 2012. URL: <https://www.raspberrypi.org/documentation/configuration/config-txt/overclocking.md> (visitado 04-09-2019).
- [18] Raspberry Pi Foundation. *Teach, Learn, and Make with Raspberry Pi*. 2012. URL: <https://www.raspberrypi.org/> (visitado 04-09-2019).
- [19] Raspberry Pi Foundation. *Raspberry frequency management and thermal control*. 2016. URL: <https://github.com/raspberrypi/documentation/blob/master/hardware/raspberrypi/frequency-management.md> (visitado 04-09-2019).
- [20] Raspberry Pi Foundation. *Raspberry Pi 3 Model B*. 2016. URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (visitado 04-09-2019).
- [21] Henry. *A Comparison of Intel's 32nm and 22nm Core i5 CPUs: Power, Voltage, Temperature, and Frequency*. Oct. de 2012. URL: <http://blog.stuffedcow.net/2012/10/intel32nm-22nm-core-i5-comparison/> (visitado 04-09-2019).
- [22] honzulas. *Intel Speed Shift technology doesn't work how expected*. 2019. URL: <https://forums.intel.com/s/question/0D50P00004DnJTMSA3/intel-speed-shift-technology-doenst-work-how-expected-i7-8850h?language=es> (visitado 04-09-2019).
- [23] Brett Howse. *Examining Intel's New Speed Shift Tech on Skylake: More Responsive Processors*. 2015. URL: <https://www.anandtech.com/show/9751/examining-intel-skylake-speed-shift-more-responsive-processors> (visitado 04-09-2019).
- [24] *HWBOT Overclocker Leagues - 2019 Season*. HWBOT classic. URL: <https://hwb0t.org/league> (visitado 04-09-2019).

- [25] HWiNFO. *HWiNFO - Free System Information, Monitoring and Diagnostics*. URL: <https://www.hwinfo.com/> (visitado 04-09-2019).
- [26] DIODES Incorporated. *APX803*. 2010. URL: <https://www.mouser.com/ds/2/115/APX803-82466.pdf> (visitado 04-09-2019).
- [27] DIODES Incorporated. *PAM2306*. 2012. URL: <https://www.diodes.com/assets/Datasheets/PAM2306.pdf> (visitado 04-09-2019).
- [28] *Intel 64 and IA-32 Architectures Software Developer's Manual*. 325384. Ver. 071. Intel® Corporation. Oct. de 2019.
- [29] *Intel Announces New 22 nm 3D Tri-gate Transistors*. Intel® Corporation. Abr. de 2011. URL: <https://www.intel.com/content/www/us/en/silicon-innovations/standards-22nm-3d-tri-gate-transistors-presentation.html> (visitado 04-09-2019).
- [30] *Intel® Core X-Series Processor Families Datasheet*. 335899. Ver. 004. Intel® Corporation. Oct. de 2018.
- [31] Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Ahmed Heman, Axel Jantsch y Hannu Tenhunen. «A Perspective on Dark Silicon». En: ene. de 2017. DOI: [10.1007/978-3-319-31596-6_1](https://doi.org/10.1007/978-3-319-31596-6_1).
- [32] Kashif Nizam Khan, Mikael Hirki, Tapio Niemi, Jukka K. Nurminen y Zhonghong Ou. «RAPL in Action: Experiences in Using RAPL for Power Measurements». En: *ACM Trans. Model. Perform. Eval. Comput. Syst.* 3.2 (mar. de 2018), 9:1-9:26. ISSN: 2376-3639. DOI: [10.1145/3177754](https://doi.org/10.1145/3177754). URL: <http://doi.acm.org/10.1145/3177754>.
- [33] Kashif Nizam Khan, Sanja Scepanovic, Tapio Niemi, Jukka K. Nurminen, Sebastian Von Alftan y Olli-Pekka Lehto. «Analyzing the power consumption behavior of a large scale data center». En: *SICS Software-Intensive Cyber-Physical Systems* 34.1 (mar. de 2019), págs. 61-70. ISSN: 2524-8529. DOI: [10.1007/s00450-018-0394-7](https://doi.org/10.1007/s00450-018-0394-7). URL: <https://doi.org/10.1007/s00450-018-0394-7>.
- [34] Alexey Kopytov. *Sysbench*. URL: <https://github.com/akopytov/sysbench> (visitado 04-09-2019).
- [35] Antony Leather. *Intel Core i7-7800X Review*. 2017. URL: <https://bit-tech.net/reviews/tech/cpus/intel-core-i7-7800x-review/6/> (visitado 17-10-2019).
- [36] Newtons4th Ltd. *N4l PPA500 Power Analyzers*. 2017. URL: <https://www.newtons4th.com/products/power-analyzers/ppa500/> (visitado 04-09-2019).
- [37] Newtons4th Ltd. *PPA5xx-15xx-User-Manual-v3.00*. 2017. URL: https://www.newtons4th.com/wp-content/uploads/2014/07/PPA5xx_15xx-User-Manual-v3.00.pdf (visitado 04-09-2019).

- [38] Steven Martin, Trevor Flautner Krisztián y Mudge y David Blaauw. «Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads». En: nov. de 2002, págs. 721-725. DOI: [10.1109/ICCAD.2002.1167611](https://doi.org/10.1109/ICCAD.2002.1167611).
- [39] MaxLinear. *MxL7704-MaxLinear*. 2018. URL: <https://www.maxlinear.com/product/power-management/universal-pmics/universal-pmics/mxl7704> (visitado 04-09-2019).
- [40] Trevor Mudge. «Power: a first-class architectural design constraint». En: *Computer* 34.4 (abr. de 2001), págs. 52-58. ISSN: 0018-9162. DOI: [10.1109/2.917539](https://doi.org/10.1109/2.917539).
- [41] Noctua. *NH U9S*. URL: <https://noctua.at/es/nh-u9s> (visitado 17-10-2019).
- [42] Overclock.net. *Overclock.net - An overclock community*. URL: <https://www.overclock.net/> (visitado 17-10-2019).
- [43] Ronak Lad Pavan H Vora. *A Review Paper on CMOS, SOI and FinFET Technology*. URL: <https://www.design-reuse.com/articles/41330/cmos-soi-finfet-technology-review-paper.html> (visitado 04-09-2019).
- [44] R Pindado. «Phase Locked-Loop (PLL): Fundamento y aplicaciones». En: ().
- [45] Murali Prasad y U. B. Mahadevaswamy. «Comparative Study of MOSFET, CMOS and FINFET: A Review». En: 2017.
- [46] International Rectifier. *IR35201 Datasheet*. 2015. URL: https://www.infineon.com/dgdl/Infineon-IR35201MTRPBF-DS-v01_00-EN.pdf?fileId=5546d462576f347501579c95d19772b5 (visitado 17-10-2019).
- [47] Seagate. *FireCuda Compute, Product Manual*. 2016. URL: <https://www.seagate.com/www-content/product-content/sshd-fam/desktop-sshd/en-us/docs/100804185a.pdf> (visitado 17-10-2019).
- [48] Tomaž Šolc. *NOTES ON THE GENERAL-PURPOSE CLOCK ON BCM2835*. 2019. URL: https://www.tablix.org/~avian/blog/archives/2018/02/notes_on_the_general_purpose_clock_on_bcm2835/ (visitado 04-09-2019).
- [49] KPC Team. *Overclocking for Raspberry Pi 3 Model B*. 2017. URL: https://xdevs.com/article/rpi3_oc/ (visitado 04-09-2019).
- [50] Kingston Technology. *KVR24N17S8/8 Memory Module Specification-Extreme*. 2016. URL: https://www.kingston.com/dataSheets/KVR24N17S8_8.pdf (visitado 17-10-2019).
- [51] Kingston Technology. *SSD PCIe NVMe A100 - 240GB*. 2016. URL: <https://www.kingston.com/es/ssd/a1000-solid-state-drive> (visitado 17-10-2019).

- [52] *The Intel Skylake-X Review: Core i9 7900X, i7 7820X and i7 7800X Tested.* Ian Cutress. Jun. de 2019. URL: <https://www.anandtech.com/show/11550/the-intel-skylakex-review-core-i9-7900x-i7-7820x-and-i7-7800x-tested/1> (visitado 04-09-2019).
- [53] Tom'sHardware. *Tom's Hardware: For The Hardcore PC Enthusiast.* URL: <https://www.tomshardware.com/> (visitado 17-10-2019).
- [54] *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1 Design Guidelines.* 321736. Ver. 002. Intel® Corporation. Sep. de 2009.
- [55] Igor Wallossek. *How to Stress-Test CPUs and PCs.* 2018. URL: <https://www.tomshardware.com/reviews/stress-test-cpu-pc-guide,5461-7.html> (visitado 04-09-2019).
- [56] Wikichip. *Skylake (server) - Microarchitectures - Intel.* 2017. URL: [https://en.wikichip.org/wiki/intel/microarchitectures/skylake_\(server\)#Entire_SoC_Overview](https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server)#Entire_SoC_Overview) (visitado 17-10-2019).
- [57] Wikipedia. *Mersenne Prime.* 2018. URL: https://en.wikipedia.org/wiki/Mersenne_prime (visitado 04-09-2019).
- [58] Rafael J. Wysocki. *CPU performance Scaling.* 2017. URL: <http://www.ti.com/lit/an/slva646/slva646.pdf?DCMP=fullyc&HQS=pwr-alps-lpdc-fullyc-fr59x69x-20140624-dvs-mcs-en> (visitado 04-09-2019).